

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені ІГОРЯ
СІКОРСЬКОГО»**

**Інститут прикладного системного аналізу
Кафедра математичних методів системного аналізу**

До захисту допущено
В. о. завідувача кафедри
_____ О.Л. Тимощук
«__» _____ 2020 р.

**Дипломна робота
на здобуття ступеня бакалавра**

**за освітньо-професійною програмою «Системи та методи штучного
інтелекту»**

**спеціальності 122 «Комп'ютерні науки та інформаційні технології»
на тему: «Порівняльний аналіз методів прогнозування фондового ринку»**

Виконав: студент IV курсу, групи КА-65

Русин Данило Костянтинович _____

Керівник: д.т.н., проф. Зайченко Олена Юріївна _____

Консультант з економічного розділу: к.е.н., доц. Шевчук О. А. _____

Консультант з нормоконтролю: к.т.н., доц. Коваленко А.Є. _____

Рецензент: к.т.н. Вішталъ Д.М. _____

Засвідчую, що у цій дипломній
роботі немає запозичень з праць
інших авторів без відповідних
посилань.

Студент _____

Київ – 2020 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Інститут прикладного системного аналізу
Кафедра математичних методів системного аналізу

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 122 «Комп’ютерні науки та інформаційні технології»

Освітньо-професійна програма «Системи та методи штучного інтелекту»

ЗАТВЕРДЖУЮ

В.о. завідувача кафедри

_____ Оксана ТИМОЩУК

«25» травня 2020 р.

ЗАВДАННЯ
на дипломну роботу студенту
Русину Данилу Костянтиновичу

1. Тема роботи «Порівняльний аналіз методів прогнозування фондового ринку», керівник роботи д.т.н., проф. Зайченко Олена Юріївна, затверджена наказом по університету від «25» травня 2020 р. № 1143-с
2. Термін подання студентом роботи 8.06.2020.
3. Вихідні дані до роботи

Вихідними даними є проведення порівняльний аналіз трьох методів прогнозування.

Вхідними даними для проведення аналізу є набір спостережень за вартістю цінних паперів конкретної компанії в конкретний період часу.

4. Зміст роботи

1. Ознайомитись з методами прогнозування фондового ринку.
2. Визначити найкращий набір вхідних даних та критерії оцінки моделі.

3. Побудувати моделі та прогнозувати динаміку розвитку на майбутній проміжок часу.
4. Програмно реалізувати проведений аналіз.
5. Перелік ілюстративного матеріалу (із зазначенням плакатів, презентацій тощо)
 - 5.1 Презентація до захисту роботи.
 - 5.1.1 Слайд «модель ARIMA»
 - 5.1.2 Слайд «Алгоритм навчання LSTM»
 - 5.1.3 Слайд «PROPHET»
 - 5.1.4 Слайд «Аналіз результатів»

6. Консультанти розділів роботи*

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Економічний	Шевчук О. А., доцент		

7. Дата видачі завдання 01.02.2020

Календарний план

№ з/п	Назва етапів виконання дипломної роботи	Строк виконання етапів роботи	Примітка
1.	Отримання завдання	01.02.2020	
2.	Збір інформації	20.02.2020	
3.	Аналіз вимог завдання, вибір методів і засобів розв'язання поставленої задачі	21.03.2020	
4.	Провести пошук існуючих методів прогнозування	22.04.2020	
5.	Вибір найоптимальніших методів, та проведення порівняння на одних даних	15.05.2020	
6.	Функціонально вартісний аналіз	22.05.2020	
7.	Загальні висновки	26.05.2020	
8.	Оформлення дипломної роботи	31.05.2020	
9.	Отримання допуску до захисту та подача роботи в ДЕК	02.06.2020	

Студент

(підпис)

Русин Д. К.

Керівник проекту (роботи)

Зайченко О. Ю.

РЕФЕРАТ

Дипломна робота: 111 с., 4 ч., 10 табл., 13 рис., 2 дод., 23 джерела.

ПРОГНОЗУВАННЯ, АНАЛІЗ ЧАСОВИХ РЯДІВ, НЕЙРОННІ МЕРЕЖІ, ФОНДОВИЙ РИНОК

Мета роботи – проаналізувати існуючі методи моделювання та прогнозування та реалізувати їх на вхідних даних.

Об'єкт дослідження – показники зразків банкнот що характеризують ступінь зношування.

У роботі проаналізовано та побудовано методи прогнозування цін на фондовому ринку. Проведено порівняння трьох методів: модель Бокса-Дженкінса, нейронна мережа Long Short-Term Memory, та інтегроване рішення Prophet. Побудовано прогноз для вартості акцій компанії Apple на майбутній період.

Розробки і розрахунки велися мовою програмування Python в середовищі розробки Jupyter Notebook.

ANNOTATION

Graduate work: 111 p., 4 p., 10 tables, 13 figures, 2 appendices, 23 sources.
FORECASTING, TIME ANALYSIS, NEURAL NETWORKS, STOCK
MARKET

The purpose of the work is to analyze the existing methods of modeling and forecasting and implement them on the input data.

The object of study - indicators of banknote samples that characterize the degree of wear.

The paper analyzes and constructs methods for forecasting stock market prices. Three methods were compared: The Box-Jenkins model, the Long Short-Term Memory neural network, and the integrated Prophet solution. The forecast for the value of Apple shares for the future is made.

Developments and calculations were conducted in the Python programming language in the Jupyter Notebook development environment.

ЗМІСТ

ВСТУП	8
РОЗДІЛ 1 ЗАГАЛЬНИЙ ОГЛЯД СВІТОВОГО ФОНДОВОГО РИНКУ	9
1.1 Основні поняття	9
1.2 Історія утворення фондових ринків	10
1.3 Чому фондовий ринок це важливо?	13
1.4 Як працюють фондові біржі?.....	14
1.5 Які біржі існують?	18
1.6 Що таке фондові індекси?.....	20
1.7 Кризові ситуації на фондовому ринку	23
1.7.1 Тюльпанова криза	23
1.7.2 Велика Депресія	24
1.7.3 Чорний понеділок.....	26
1.7.4 Криза доткомів	26
1.7.5 Іпотечна криза 2008. Велика рецесія	28
1.8 Постулати технічного аналізу	30
1.8.1 Ринок враховує все.....	30
1.8.2 Рух цін підпорядкований тенденціям	32
1.8.3 Історія повторюється	33
1.9 В чому різниця між Технічним і Фундаментальним аналізом?...	33
1.10 Висновки до розділу	37
РОЗДІЛ 2 МАТЕМАТИЧНІ МЕТОДИ ТА НАПРЯМКИ ПРОГНОЗУВАННЯ ДИНАМІКИ ФОНДОВИХ РИНКІВ	38

2.1 Математичні моделі часових рядів	38
2.2 Модель Бокса-Дженкінса ARIMA.....	40
2.3 Методи нейронних мереж.....	44
2.4 LSTM нейронні мережі	46
2.5 Методологія програмного пакету Prophet.....	48
РОЗДІЛ 3 ПОРІВНЯЛЬНИЙ АНАЛІЗ РЕЗУЛЬТАТІВ МОДЕЛЮВАННЯ ТА	
ПРОГНОЗУВАННЯ ДИНАМІКИ ФОНДОВОГО РИНКУ	51
3.1 Прогнозування.....	51
3.2 Методи оцінки параметрів моделі	52
3.3 Вхідні дані	53
3.4 Порівняльний аналіз моделей прогнозування	55
3.5 Висновки до розділу	63
РОЗДІЛ 4 ФУНКЦІОНАЛЬНО-ВАРТІСНИЙ АНАЛІЗ	64
4.1 Постановка завдання проектування	64
4.2 Обґрунтування функцій та параметрів програмного продукту ..	64
4.3 Економічний аналіз варіантів розробки ПП	73
4.4 Висновки до розділу 4	79
ВИСНОВКИ.....	81
ДОДАТОК А ІЛЮСТРАТИВНИЙ МАТЕРІАЛ.....	84
ДОДАТОК Б ЛІСТИНГ ПРОГРАМИ	95

ВСТУП

В новітній історії людства, найдосконалішим способом фінансових стосунків між людьми в сучасній капіталістичній економіці став фондовий ринок. Це поняття, та звіт правил що його регулюють дозволяють одним людям ставати мільйонерами та мільярдерами, іншим залучати інвестиції для своїх компаній з різних точок світу, а деяким лишитись усього та набуту купу боргів. Щоб досягнути фінансових результатів, необхідно розуміти усі правила гри та вміти аналітично оцінювати ситуацію. Зараз, на початок 2020 року капіталізація світового ринку цінних паперів досягає 70 трильйонів доларів, а великі гравці кожен день укладають угоди на мільярди доларів. Але це зовсім не означає що звичайній людині з початковим капіталом в 100\$ нема чого робити в цьому океані з акулами бізнесу. Для успішного фінансового плану необхідно вміти правильно аналізувати ситуацію, шукати кореляції між економічними чи політичними подіями та вартістю акцій, прогнозувати ріст ціни на деякий період вперед, правильно скласти hold-sell-buy стратегії.

Данна робота створена для того, щоб дослідити сучасні методи технічного аналізу, та зрівняти основні методи прогнозування цін акцій. Тут фондовий ринок розглядатиметься лише на макроекономічному рівні, як складна саморегулююча система.

Актуальність роботи особливо виражена в сучасній кризовій ситуації, коли світ зіштовхнувся зі світовою пандемією COVID-19. Ця ситуація стала новим викликом для світової економіки, і ні в кого немає повного розуміння як ситуація зміниться в майбутньому.

Тож проведемо технічний аналіз ринку та перевіримо роботу методів математичного прогнозування на базі штучних нейронних мереж в різних ситуаціях.

РОЗДІЛ 1 ЗАГАЛЬНИЙ ОГЛЯД СВІТОВОГО ФОНДОВОГО РИНКУ

1.1 Основні поняття

Нижче наведені основні терміни та поняття які будуть використовуватись в даній роботі. Визначення взяті з Закону України про цінні папери та фондовий ринок[1].

Фондовий ринок або ринок цінних паперів - сукупність учасників фондового ринку та правовідносин між ними щодо розміщення, обігу та обліку цінних паперів.

Цінним папером є документ установленої форми з відповідними реквізитами, що посвідчує грошове або інше майнове право, визначає взаємовідносини емітента цінного папера (особи, яка видала цінний папір) і особи, що має права на цінний папір, та передбачає виконання зобов'язань за таким цінним папером, а також можливість передачі прав на цінний папір та прав за цінним папером іншим особам.

Емітент - юридична особа, у тому числі Фонд гарантування вкладів фізичних осіб або міська рада, а також держава в особі уповноважених нею органів державної влади чи міжнародна фінансова організація, які від свого імені розміщують емісійні цінні папери та беруть на себе зобов'язання за ними перед їх власниками.

Акція – іменний цінний папір, який посвідчує майнові права його власника (акціонера), що стосуються акціонерного товариства, включаючи право на отримання частини прибутку акціонерного товариства у вигляді дивідендів та право на отримання частини майна акціонерного товариства у разі його ліквідації, право на управління акціонерним товариством, а також немайнові

права, передбачені Цивільним кодексом України та законом, що регулює питання створення, діяльності та припинення акціонерних товариств.

Дивіденд – платіж, що проводиться юридичною особою на користь власників (довірених осіб власника) корпоративних прав, емітованих такою юридичною особою, у зв'язку з розподілом частини її прибутку[2].

Облігація – емісійний цінний папір, що засвідчує внесення її власником коштів і підтверджує зобов'язання відшкодувати йому номінальну вартість цього цінного паперу, в передбачений у ньому строк з виплатою фіксованого відсотка (якщо інше не передбачено умовами випуску)[2].

Бики і Ведмеді - такими загальноприйнятими термінами називають учасників ринку цінних паперів. Бичій та Ведмежий ринок асоціюються з ростучим та падаючим ринком відповідно. Тож биками називають трейдерів-оптимістів, що роблять свій основний капітал на рості ринку, в ведмежми- тих, хто заробляє на падінні.

1.2 Історія утворення фондових ринків

Перші прототипи сучасних угод з цінними паперами можна знайти ще в часи Римської імперії, коли інвестори вкладали свої гроші в військові кампанії, в обмін на дивіденди в вигляді награваного майна, або ж фінансували віддалені провінції. Також з записів Цицерона збереглися згадки про безготівкові фінансові транзакції, через записи в банківських бухгалтерських книгах, угоди з іпотечними цінними паперами та про приватні компанії що допомагали з проведенням фінансових переводів в будь яку точку імперії.

Наступним кроком розвитку можна назвати епоху великих географічних відкриттів. Будь яка людина могла купити державні облігації, фінансування морських походів в Америку, Індію, Африку, в обмін на привезені назад цінності.

У 1602 р. Була створена перша в світі офіційна фондова біржа, Амстердамська фондова біржа. На початку вона була створена для торгівлі акціями Голландської Ост-Індської компанії, першою приватною компанією що випускала корпоративні облігації та акції для всіх охочих. Це вважається першим прикладом того, що можна назвати схожим на сучасні фондові ринки, з такими атрибутами, як торгівля акціями, отримання дивідендів, правила торгівлі на біржі, стратегії гри на біржі для отримання прибутку. В цей період також було створено першу книгу, написану про торги на фондовій біржі, у 1688 році під назвою «Збентеження конфузій», написану Джозефом де ла Вега, успішним торговцем акціями того часу. Він розповів про роботу амстердамського фондового ринку та дав загальні поради щоб досягти успіху на цьому ділі.

Все це стало стало можливим завдяки вміло створеній правовій системі фондового ринку, та незалежній судовій гілці. В суді кожен міг відстояти свої права, що й приваблювало інвесторів саме до Амстердаму.

Після успіху Амстердамського фондового ринку ідея створення інших фондових ринків розповсюдилась по всій Європі, хоч і минуло маже століття перш ніж було створено конкурентоспроможну структуру. В кінці 17 століття англійський король Вільгельм III шукав спосіб залучення додаткових грошей для ведення війн, і почав випускати державні облігації, що призвело до створення Банку Англії. Це сприяло розповсюдження інтересу до торгівлі ними, тож приватні компанії вирішили випускати свої акції для усіх охочих. Тож це потребувало створенню механізму покупки-продажі акцій на вторинному ринку. Основним місцем для проведення цієї діяльності в Лондоні була Королівська біржа, яка уже на той час була фінансовим центром в місті більше ста років. Зі збільшенням кількості бажаючих торгувати акціями, ріс і поріг входу на Королівську біржу. Тому усе більше людей проводили угоди у прилеглих кав'ярнях, і незабаром цей район почав називатися Біржовою алеєю. Ці кав'ярні

розміщували ціни на цінні папери на своїх дошках оголошень, а інвестори відвідували їх для торгівлі акціями. Тож вони стали провісниками перших великих фондових компаній. Одна з них, кав'ярня Lloyd's, перетвориться на London Lloyd's, одну з найбільших у світі страхових компаній.

Паралельно з розвитком Лондонської біржі, інші країни Європи та США також почали розвиток своїх фондових ринків. Саме до США біржі потрапили в 1790 році в Філадельфію, через представництва англійських компаній.

Сьогодні для більшості людей назва Уолл-Стріт є синонімом біржі. Ринок на Уолл-стріт відкрився 17 травня 1792 року на розі Уолл-стріт та Бродвею. Двадцять чотири біржові посередники підписали Баттонвудський договір про організацію торгівлі цінними паперами. Що послугувало створенню Нью-Йоркської фондової та біржової комісії в 1817 році. Згодом її називали просто Нью-Йоркською фондовою біржою або NYSE. Сьогодні NYSE - це найбільша біржа в світі, яка має близько 20 трильйонів доларів ринкової капіталізації.[4]

На початку 1990х почали набирати популярність Інтернет біржі, або як їх називають трейдингові платформи. Це сворило новий період в фондових операціях. Онлайн платформи були спеціально розроблені, щоб люди могли отримати доступ до фінансових ринків, до яких раніше могли звертатися лише спеціалізовані торгові фірми. Тепер будь хто може проводити торгові операції зі свого смартфона з будь якої точки планети.

NASDAQ, на даний момент друга за величиною фондова біржа у світі, повністю базується в мережі Інтернет.

1.3 Чому фондовий ринок це важливо?

Історія показала, що вартість цінних паперів є важливою частиною динаміки економічного розвитку і може впливати або бути показником соціального, економічного та політичного настрою. Економіка, де фондовий ринок зростає, вважається економікою, що розвивається. Фондовий ринок часто вважається основним показником економічної міцності та розвитку країни. Підвищення цін на акції, наприклад, пов'язане із збільшенням інвестицій у бізнес та навпаки. Ціни на акції також впливають на багатство домогосподарств та їх споживання.

Крім того фондовий ринок займає мабуть найважливішу роль в функціонуванні світової економіки та виконує ряд функцій.

По-перше фондовий ринок це основний інструмент для фінансування компаній, що виставляють свої акції для обігу на фондових біржах. Зазвичай це дозволяє компаніям розвиватись набагато швидше, ніж якщо б вони фінансували себе з прибутку. Без цього можливо не існувало б великих транснаціональних компаній та інноваційних стартапів, а отримати гроші на розвиток було б набагато важче.

А для інвесторів це привабливий спосіб заробітку, адже вклавши гроші в правильну компанію і в правильний момент, можна в майбутньому примножити свій капітал в багато разів. Ліквідність на фондовому ринку набагато більша ніж на ринку товарів чи ринку нерухомості, що спонукає фінансуванню саме в цінні папери.

По-друге ринок цінних паперів це потужний інструмент для перерозподілу капіталу:

- Між країнами та територіями.
- Між галузями промисловості і секторами економіки.
- Між окремими підприємствами всередині одного сектора.

Необхідно розуміти, що ринок влаштований таким чином, що гроші перетікають туди, де можуть принести найбільший ефект від свого використання. Цей принцип проявляється при перерозподілі капіталу завжди і всюди - неважливо, чи йде мова про приватні компанії або цілі країни. Наприклад купляючи акції української компанії, інвестор тим самим інвестує і в економіку України. А інвестиції в одну компанію з фармацевтичного сектору допомагають в конкурентній боротьбі з іншими компаніями цього сектору.

З іншої ж сторони завдяки цінним паперам здійснюється перерозподіл інвестицій в кожне конкретне підприємство між великими, середніми і малими інвесторами. Цей процес стосується майже будь-якого громадянина країни, навіть якщо той про це не підозрює. Так, умовна людина, яка має вклад у банку, може не знати і не думати про те, що банк, використовуючи його гроші, міг купити, наприклад, корпоративні облігації - саме так, конкретна людина, не підозрюючи про це, стає джерелом коштів розвитку конкретного підприємства і економіки в цілому.

1.4 Як працюють фондові біржі?

Зазвичай люди уявляють фондову біржу як велике, технологічно оснащене приміщення з сотнями моніторів, де імпозантні чоловіки в дорогих костюмах із шкіряними дипломатами роблять гроші з повітря. [4-7]

Найперше що варто написати- фондові біржі, в звичному для нас розумінні це фактично вторинний ринок цінних паперів, що працює за моделлю аукціону. Ще декілька десятиків років тому торгівля на біржі виглядала як очний аукціон, де трейдери голосно перекрикуючи один одного виголошували суму, яку вони згодні віддати за акцію. Зараз же в ХХІ столітті такий спосіб комунікації між

покупцем і продавцем замінили електронні системи. Але для того щоб розібратись як же воно працює, необхідно розглянути ключові ролі та процеси, що існують на біржах.

Інвестор- так називають людину у якої є капітал, та бажання збільшити його шляхом торгів на біржі. Що цікаво, як написано вище, інвестором може бути особа, яка навіть не підозрює що вона опосередковано інвестором. Наприклад, особа що поклала гроші на депозит до банку.

В загальному, кожен інвестор який хоче торгувати на біржі, має заключити контракт з посередником, що надає брокерські послуги. Брокерами можуть бути лише ті особи, що мають спеціальні ліцензії для заключення біржових угод. Ніхто окрім ліцензованих брокерів не має права торгувати на біржі.

Щоб успішно вести фінансову діяльність на біржі інвестор має вибрати стратегію гри.

Брокери- це професійні посередники між інвестором і біржою. Основні задачі брокерів це:

- Вести торгову діяльність від імені свого замовника;
- Зводити покупця з продавцем, та продавця з покупцем акцій;
- Прораховувати ризики, розроблювати стратегії, створювати портфелі акцій;

Зазвичай за свої послуги брокери беруть відсоток від заробленої виручки, але також можуть брати і фіксовану ставку. Найбільші брокерські компанії залучають капітал на мільйони та мільярди доларів США.

Торгова система біржі, це електронний інструмент, площадка де і відбувається основна частина торгової діяльності. Брокери залишають в торговій системі свої заявки на купівлю чи продажу акцій за певну ціну, і програмним алгоритмом заявка покупця знаходить відповідну заявку продавця, та навпаки.

Такі угоди відправляються на наступний крок, реєстрацію. В інтернет біржах така система називається Торговою платформою.

Клірингові компанії надають послуги реєстраторів угод. В основні функції клірингових компаній також входить:

- Перевіряти на коректність та легітимність угоди;
- Реєструвати угоду;
- Гарантувати виконання угоди обома сторонами;
- Забезпечити доставку акцій покупцеві та грошей продавцеві;

Окрім вище описаних ролей, існують організації, що відповідають за усе юридичне та політичне урегулювання фондового ринку. **Регулятори** це державні установи, які узаконюють заключення угод, випуск акцій, надають ліцензії брокерам. Найбільш впливові регулюючі органи в світі це Національна ф'ючерсна асоціація США (NFA), та Комісія з термінової біржової торгівлі (CFTC). В Україні цю задачу виконує Національна комісія з цінних паперів та фондового ринку (НКЦПФР).

Схема роботи фондових бірж наведена на рисунку 1.1.

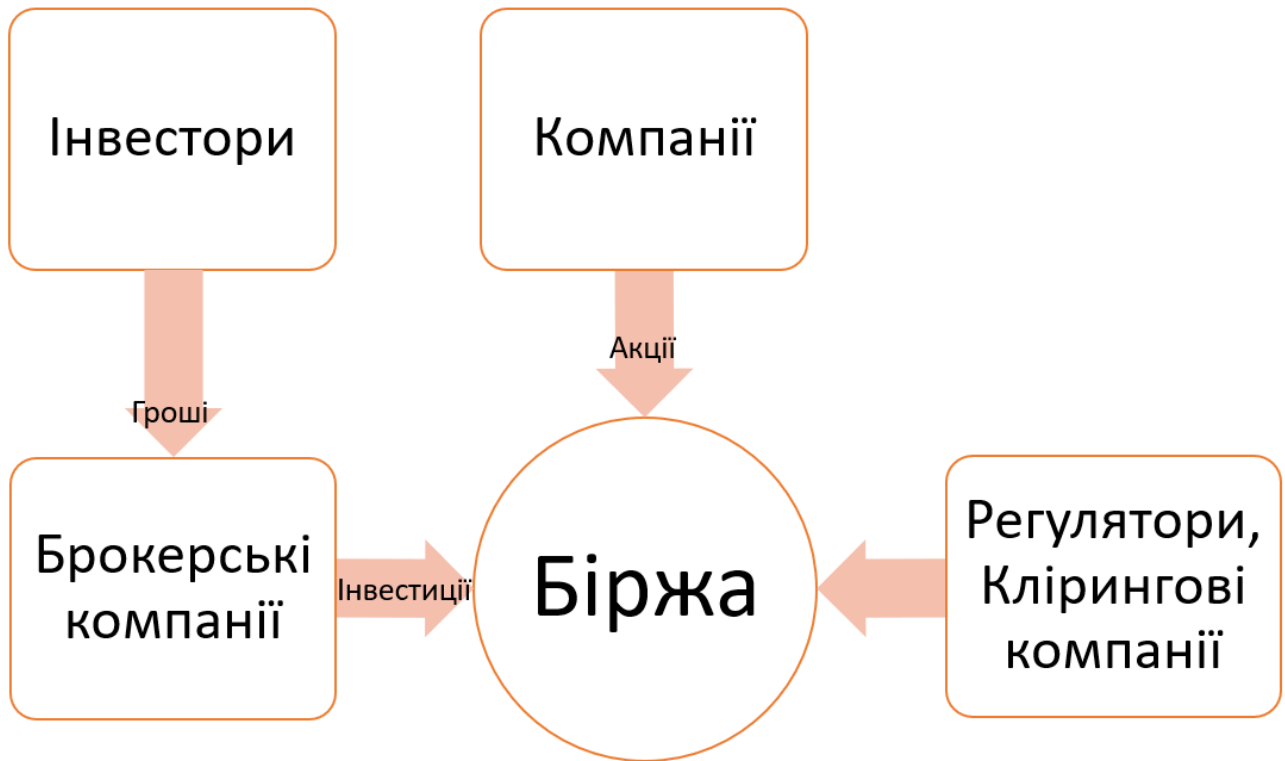


Рис. 1.1 — Схема роботи фондових бірж

Отже, тепер можна навести приклад як відбуваються біржові операції.

Інвестор, що хоче купити акцію повідомляє про свої наміри брокерську компанію. Брокерська компанія передає завдання своєму представнику, що знаходиться на відповідній біржі. Представник вносить заявку з інформацією які цінні папери він хоче купити та за яку ціну. Його заявка «знаходить» відповідну заявку, що відповідає бажанню продати акцію за ту ж ціну. Угода реєструється в протокольному списку, і клірингова компанія допомагає обом учасникам провести обмін згідно закону. Інформація про покупку повертається тим же маршрутом до інвестора.

Вище описаний алгоритм торгівлі на класичній біржі. Але також існують інтернет біржі, де дії учасників можуть кардинально відрізнятись. [9,20]

1.5 Які біржі існують?

Майже в кожній розвинутій країні є своя біржа, а в деяких країн навіть більше однієї. Історично склалось, що дві найбільші фондові біржі в світі знаходяться в США, а саме в Нью-Йорку. На Нью-Йоркській фондовій біржі торгують акціями понад 5000 компаній, а загальна ринкова капіталізація перевищує 35 трильйон доларів. Окрім американських бірж, вагомий вплив мають біржі Східної Азії та Європейського союзу. Нижче наведено інформацію про 5 найбільших бірж світу.

1. New York Stock Exchange (NYSE) – Нью-Йоркська фондова біржа почала свою історію в 1792 році з підписання Баттонсудської угоди, і з тих пір займає першість в світовій економіці. Уся інформація що глибоко укорінилось в свідомості людей і асоціюється з Уолл-Стріт пов'язана саме з NYSE. Біржа використовує два способи торгівлі: напряму між брокерами та через електронну систему. Не залежно від способу усі торги проходять за правилами аукціону. На NYSE працюють найпрофесійніші брокери з найвищою категорією ліцензії. На біржі торгують акціями американських та всесвітніх компаній, що виставляють в основному сильні та надійні акції. Нью-Йоркська біржа визначає промисловий індекс Доу-Джонса, індекси NYSE ARCA Tech 100 Index, та NYSE Composite. Про біржові індекси буде написано в наступній частині.

2. NASDAQ (National Association of Securities Dealers Automated Quotation) - Національна асоціація автоматизованих котирувань цінних паперів, друга по об'єму капіталу біржа в США, та перша в світі біржа, що стала торгувати через електронну систему в інтернеті. NASDAQ запустив свою біржу ще в 1971 році, ще коли комп'ютери були малопродуктивними а інтернет тільки почав зароджуватись, згодом NASDAQ стала першою біржою що веде торги через веб-сайт. Зараз NASDAQ торгує акціями найтехнологічніших компаній світу, таких

як Apple, Amazon, Microsoft, Facebook, Gilead Sciences, Starbucks, Tesla, Intel та Oracle. Варто зазначити що угоди на цій біржі укладаються в рази скоріше, а кількість таких угод перевищує кількість на Нью-Йоркській біржі, так як в 80% випадків акціями торгують спеціальні роботи.

3. В результаті об'єднання двох найбільших бірж Японії (Осакійської біржі цінних паперів та Токійської фондової біржі) утворилася третя за величиною торгова площадка в світі - Japan Exchange Group. Офіційною датою відкриття JPX є 1 січня 2013 року, але Токійська та Осакійська біржі почали функціонувати ще в 1878р.

Капіталізація ринку на червень 2018 року становила понад \$ 6 трлн. Лістинг налічує 3 628 компаній, а обсяги торгів в середньому тримаються близько \$ 0,5 трлн.

4. Найбільша торговельна платформа континентального Китаю, Shanghai Stock Exchange, почала торги в 1860 роки. SSE є некомерційною організацією, що управляється Комісією з регулювання цінних паперів Китаю. Відмінною рисою біржі є торгівля двома типами акцій. Перший тип торгується за юані і доступний всім інвесторам. Другий тип номінований в доларах США і спочатку був доступний тільки нерезидентам.

З 2014 року обмеження були зняті, що дало свої плоди. З того моменту капіталізація ринку зросла майже в 2 рази до \$ 4,5 трлн., А обороти за 4 роки піросли до \$ 470 млрд.

5. Euronext в своєму становленні найбільшою біржою Європи пішла по шляху активних злиттів і поглинань. Народження площадки відбулося в 2000 році в результаті злиття Паризької, Брюссельської та найстарішої в світі Амстердамської біржі. Пізніше до них приєдналася Лісабонська фондова біржа. Через 4 роки в 2006 році об'єднана платформа перейшла під контроль NYSE

Group, утворивши NYSE Euronext, яка в 2013 році була викуплена холдингом ICE, сформувавши найбільшу глобальну торговельну площадку.

Euronext була виділена в окрему компанію, капіталізація ринку якої становить \$ 4,3 трлн, щоденний оборот - \$ 200 млрд.

1.6 Що таке фондові індекси?

Фондовий Індекс це пакет акцій, зібраний таким чином, щоб відобразити вартісну інформацію про конкретний ринок, сектор, країну, групу компаній, тощо. Індекси можуть бути широко направленими або вузько направленими. Наприклад, NASDAQ Composite відстежує тисячі звичайних акцій та цінних паперів, якими торгують на біржі NASDAQ. Більш вузько направлений індекс Nasdaq 100 (NDX) - це індекс, який відображає 100 найбільших нефінансових компаній, що входять до списку NASDAQ Composite.

Перший фондовий індекс був розроблений 3 липня 1884 в США журналістом газети Wall Street Journal, відомим фінансистом, засновником компанії Dow Jones & Company Чарльзом Доу. Індекс Dow Jones Transportation Average розраховувався по 11 найбільшим транспортним компаніям США. На сьогоднішній день в нього входять 20 компаній- вантажних перевізників. Однак найбільшу популярність здобув Dow Jones Industrial Average (DJIA), що розраховується по 30 найбільшим компаніям промисловості з 1928 року. Зараз же слово «промисловість» в назві індексу відображає скоріше дань історії, адже в цей список входять акції таких компаній як: Apple, Microsoft, IBM, Nike, Coca Cola Company, Visa, Boeing, Walmart, McDonald's, тощо.

Також існує декілька основних фондових індексів та сімейств фондових індексів, з якими знайомий кожен інвестор. Дійсно, деякі укладачі індексів не

обмежуються одним індексом, а створюють ціле сімейство подібних інструментів. Отже, нижче наведено приклади найпопулярніших індексів:

- Одним з найпопулярніших індикаторів є індекси, що розраховуються рейтинговим агентством Standard and Poor's (S&P). У їх число входить головний індекс S&P 500, що об'єднує 500 найбільш капіталізованих компаній США (80% торгуються на NYSE, 20% - на AMEX).
- Своім сімейством індексів володіє біржа NASDAQ - ці індикатори враховують поведінку більше 4500 американських і зарубіжних компаній. В їх число входять зведений індекс NASDAQ Composite (об'єднує всі компанії в лістингу біржі), NASDAQ National Market (компанії з США), а також маса галузевих індексів.
- В Європі помітним сімейством індексів є індикатори німецького фондового ринку (DAX 30, Dax Price Index, DAX 100 Sector Indexes і т.п.), французькі CAC, що відрізняються за кількістю вхідних в них акцій (40, 80, 120, тощо).

Як же створюються фондові індекси?

Існує декілька способів розрахунку фондових індексів:

- 1) Індекс зважений за ціною є сумою цін усіх активів, що входять в індекс, поділеної на дільник. Найвідомішим прикладом є індекс Доу Джонса:

$$DJIA = \frac{\sum P_i}{D}, \quad (1.1)$$

де P_i – вартість i -тої акції, а D – дільник індексу.[21]

Дільник змінюється для збереження безперервності значення індексу при додаванні або виключенні компаній з індексу, а також при інших корпоративних подіях (наприклад, зміна кількості акцій даної компанії,

включеної в індекс). Значення дільника публікується в The Wall Street Journal, на початок 2020 року становив $D = 0.14734071$. Даний метод є найпростішим для обчислення. Його недоліком є те, що в ньому вага кожної акції пропорційна її ціні, яка є довільною великою. В даний час за цим методом розраховуються традиційні індекси Доу Джонс, Nikkei 225. Як правило, сучасні індекси не використовують зважування за ціною.

- 2) Індекс зважений по вільній ринковій капіталізації. Більшість сучасних фондових індексів є зваженими по вільній ринковій капіталізації. Найвідоміший приклад - S&P 500. Значення індексу дорівнює сумарній ринковій капіталізації компаній поділеної на дільник.

$$Index = \frac{\sum P_i N_i}{D}, \quad (1.2)$$

P_i – ціна акції, N_i – кількість акцій на ринку. Дільник вибирається так, щоб на момент історичного початку розрахунку індексу (базова дата) його значення дорівнювало якомусь зручному числу (базового значення); наприклад для S&P 500 базове значення дорівнювало 10. Надалі, так само як і у всіх інших типах індексів, дільник змінюється для збереження безперервності значення індексу при корпоративні події. Наприклад для індексу S&P 500 на початок 2020 року дільник $D = 8921.33269716$. [3,8,22]

Фактично, фондові індекси є не лише показником зміни фондового ринку, але й усієї світової економіки. За допомогою часових графіків індексів можна побачити як змінювались очікування людей на розвиток економіки. Коли трапляється якась політична, соціальна, економічна подія, відповідний сектор економіки обов'язково на це відреагує ростом чи падінням вартості. Отже індекс, це той індикатор, що варто аналізувати в першу чергу, при знайомстві з ринком.

1.7 Кризові ситуації на фондовому ринку

На жаль, фондовий ринок це не лише постійний ріст та процвітання. Існують ситуації, коли фондовий ринок не лише зменшує капітал інвесторів, а навіть й знищує економіки і життя цілих країн. Такі явища як Велика депресія чи криза 2008 року добре відображають, як халатність та незнання звичайних офісних працівників та брокерів можуть знищувати мільярдні компанії, та «відправляти» тисячі людей на вулицю. Людей які навіть ніколи не купляли цінні папери. Далі розберемо чому, як, коли відбувались такі події.[9]

1.7.1 Тюльпанова криза

Історично склалось, що перша в світі фондова біржа була створена в Голландії, і найперший крах фондового ринку теж відбувся саме на Голландській біржі.

Завезений в 1634 році з Османської імперії в Європу тюльпан, викликав не аби який ажіотаж у Голландської інтелігенції. Саме ця квітка стала символом статусності та багатства. Ціни на неї росли з 1634 до 1637 та сягали дуже значних сум, і незабаром спекулянти з вищого та середнього класу почали закладати майно, аби придбати тюльпанові цибулі, та перепродати їх, коли ціни піднімуться ще вище. Але, як не прозаїчно, одного дня тюльпани стали нікому не потрібні. Попит на тюльпани впав до нуля, а величезні капітали тюльпанних інвесторів перетворились лише в звичайні цибулини.

Вибух цієї бульбашки привів економіку Голландії в кризову ситуацію, і знадобились роки, щоб оговтатись від неї.

Цей приклад прекрасно показує як незнання та нерегульованість ринку відображається на житті усієї країни.

1.7.2 Велика Депресія

1920-ті роки згадуються в історії США як один з найпродуктивніших періодів в економічному та соціальному житті. «Ревучі двадцяті», як називають цей період зараз, принесли з собою нечуваний ріст загального багатства нації в два рази. Будуються хмарочоси, автомобілі продаються як гарячі пиріжки, населення без сумнівів тратить гроші на розваги. В той же час в Нью-Йорку усі, від мільйонерів до прибиральників купляють акції, заробляючи великі суми. Варто зазначити, що в ті часи ВВП США становив 50% всесвітнього ВВП.

Так продовжувалось до літа 1929-го року, коли попит населення став падати, а кількість непроданих товарів почало накопичуватись, безробіття росло. Крім того, заробітна плата на той час була низькою, споживча заборгованість зростала, аграрний сектор економіки боровся через посуху та падіння цін на продовольство, а банки мали надлишки великих кредитів, які неможливо було ліквідувати. Тим не менше, ціни на акції продовжували зростати і до осені того ж року досягли стратосферних рівнів, які не можна було виправдати очікуваними майбутніми прибутками.

24-го жовтня 1929 року фінансова бульбашка лопла. Рекордна кількість акцій – 12,9 млн, була продана у цей день, відомий як «Чорний четвер». Через п'ять днів, відбувся «чорний вівторок» – близько 16 млн акцій було продано після того, як інша хвиля паніки охопила Уолл-стріт. Мільйони акцій практично обезцінились і десятки тисяч людей збанкрутували.

Обвал фондового ринку призвів до втрати довіри до фінансових установ, що ознаменувало падіння інвестицій у промисловість та будівництво. Через звільнення, американці масово втрачають роботу. А тим, кому пощастило залишитися на робочому місці, знижують заробітну платню, що ще більше

посилює падіння купівельної спроможності населення. Багато американців були змушені купувати в кредит товари першої необхідності – їжу, ліку тощо.

До 1933 року ситуація тільки погіршувалась. Фермери через посуху та падіння цін на продовольство ще у 20-ті роки, не могли дозволити собі зібрати урожай, і були змушені залишити його гнити в полях, а в цей самий час, в інших місцях люди просто голодували. Безробіття зросло до 15 мільйонів американців, що більше 20 відсотків усього населення США. Банки, що не могли повернути свої кредити, перестають видавати населенню нові. Усе це погіршувалось безграмотними діями адміністрації президента Герберта Гувера, який вважав, що уряд не повинен безпосередньо втручатися в економіку, і що влада не зобов'язана створювати нові робочі місця або забезпечити економічну допомогу для своїх громадян.

Усе почало мінятись з приходом нового президента Франкліна Рузвельта, який негайно розпочав реформаторську діяльність, для підтримки економіки і населення. Створювались нові робочі місця, нові комісії по регулюванню економіки, дешеві кредити та поповнення ринку грішми. ВВП почав повільно зростати, економіка відновлювалась.

Велика депресія завершилась з початком другої світової війни, в 1939, коли промисловість почала масово працювати для військових цілей, а безробіття подолав національний призов до армії.

Наслідки кризи були вражаючими. Велика депресія, що перекинулась частково на західну Європу, сприяла до поширення екстремістських політичних рухів у різних європейських країнах, головним чином, сприяла приходу до влади нацистського режиму Адольфа Гітлера у Німеччині.

Було створено велику кількість регулюючих інстанцій, влада навчилась правильно реагувати на кризові ситуації- створювати робочі місця, та виділяти «дешеві» гроші.

Велика депресія має величезне значення для історії світової економіки.

1.7.3 Чорний понеділок

19 жовтня 1987 року відбулось найбільше падіння індексу Доу-Джонса за один день, він впав на 23%. Падіння ринку почалось з Гонконгу, і перейшовши на Європу та США. Економіка США втратила 1.5\$ трильйонів, а капіталізація Гонконгського, Австралійського ринків впала більше ніж на 40%.

При чому, катастрофі не передували ніякі важливі події чи новини, явних причин для обвалу не було. До сих пір, не вщухають суперечки про першопричини такої події. Серед популярних версій - перевантаження технологічної інфраструктури біржі через активне використання торгових роботів, схлопування бульбашки на американському ринку (в 1985-1987 роках індекс Dow Jones виріс в два рази), а також масові розпродажі на тлі зростання геополітичної напруженості між США і Іраном.

Держоргани відреагували швидко: федеральний резерв понизив відсоткову ставку, та вплив в ринок додаткову доларову ліквідність. Біржі, в свою чергу, розробили механізм закриття торгів, при падінні індексу Dow Jones на 10%. Криза «Чорного понеділка» була погашена в зародку, і катастрофи не трапилось.

Ця подія вчить, що ринок реагує навіть на непомітні зміни в соціальній, політичній, економічній сфері, і така ситуація може трапитись в будь який день.

1.7.4 Криза доткомів

В кінці 90-х, початку 2000-х, інтернет захопив розум людей. Інвестори побачили купу нових можливостей, що несе з собою інтернет, і почали вкладати нереальні гроші. Слова «інтернет» та «онлайн» збільшували вартість компаній в

рази. А власники компаній замість створення бізнес-плану, владали гроші в маркетинг, для залучення ще більшої кількості інвестицій, та тратили на власні потреби.

10 березня 2000 року відбулось падіння індексу високотехнічних компаній NASDAQ Composite. Усього за рік індекс впав майже в 5 разів, а тисячі ІТ компаній збанкрутували разом з інвесторами. На довгі роки утворилась стійка недовіра до усіх інтернет-бізнесів.

Результатом стала хвиля звільнень та впроваджених кримінальних справ, про розтрату інвестиційних коштів. При цьому три компанії вижили, і процвітають до сих пір, це Amazon, Google, eBay.

Називають декілька першопричин утворення цієї бульбашки:

- На початку 2000х інтернет являв собою скоріше «сирим» майданчиком для спілкування, ніж серйозним інструментом. Більшість підприємців просто не знала як його можна монетизувати.
- Так як, в основному, тогочасні ІТ компанії являли собою декілька десятків співробітників, невеличкий офіс з комп'ютерною технікою та ресурс в інтернеті, то інвестори просто не знали як правильно можна оцінити акції таких компаній, тому іноді ціни доходили до божевільних сум.
- Дотками керували як правило програмісти та ІТ-генії, які нічого не розуміли ні в бізнесі, ні в мистецтві продавати свої товари, ні в монетизації.
- Ведення бізнесу за допомогою інтернету є лише інструментом для здійснення бізнес-процесу, але не самостійним бізнес-процесом.

Отже, після кризи доткомів вхідна планка для входу в ІТ-бізнес значно піднялась, саме тому ми маємо в інтернеті дійсно хороші ресурси та сайти, без яких ми не можемо уявити інтернет.

1.7.5 Іпотечна криза 2008. Велика рецесія

З 2003 року економіка США починає швидко рости, а разом з нею росте і капітал звичайного середнього американця. З ростом рівня життя в Америці з'явилися нові фінансові механізми. Одним з таких механізмів стало вкладення в нерухомість. Ціни на американську нерухомість росли із завидною стабільністю. Це можна пояснити. Так буває у всьому світі. Чим швидше росте ціна, тим сильніше підвищується попит.

При наявності дешевих грошей в Америці стали видавати «погані» кредити під заставу цієї нерухомості. Тобто кредити стали дешевими і давати їх стали всім підряд, навіть тим, кого раніше і близько до кредитів не підпускали.

Логіка банкірів зрозуміла. Поки позичальник платить - добре. Перестає платити - забирають нерухомість. А з урахуванням гарантованого зростання її вартості, банк в будь-якому випадку буде в плюсі - оплачені за кредитом гроші позичальникові не повертаються, а будинок продається за ринковою вартістю, яка виростає дуже значно з дати видачі іпотеки. Очевидно, що гроші, виручені з продажу нерухомості, повністю надходять в банк.

Далі банки роблять ще більш ризикований крок. Беруть портфель іпотечних кредитів, так званий дериватив, та продають цей цінний папір на біржовому ринку. Причому, як доводять судові розслідування, підкуплені спеціальні рейтингові агентства ставили найвищий рейтинг надійності цим паперам. Інвестори охоче купляють деривативи, адже його вартість опирається на вартість заставленого житла, а банк платить дивіденди по цьому цінному паперу з іпотечних платежів.

Таким чином, в Америці масово формувалася бульбашка нічим не забезпеченої іпотеки. Далі, за законом жанру, завжди відбувається насичення

ринку. Так буває завжди і у всьому. Коли відбувається бурхливе зростання, рано чи пізно ринок насититься і тоді з'ясується, що товар занадто переоцінений.

В силу досить високого рівня життя і масовості «поганих» кредитів, на ринку нерухомості США, почалася паніка. Безліч компаній, що займаються видачею «поганих» кредитів, стали скидати вже цей неліквід, в результаті чого ціни на нерухомість буквально звалилися. Так лопнув іпотечна бульбашка в США. Фінансово постраждало багато людей не тільки в Америці, але і далеко за її межами. Так що люди, постраждали найбільші банки зі світовим ім'ям і багатою історією. Так в Штатах сумно закінчилося іпотечне «безумство».

Отже іпотечна криза 2007 року, завдяки махінаціям банку з деривативами, перетворилась в всесвітню фінансову кризу 2008, яку економісти називають «Великою рецесією».

П'ять провідних інвестиційних банків США припинили своє існування в колишній спосіб: Bear Stearns був перепроданий, Lehman Brothers збанкрутував, Merrill Lynch перепроданий, Goldman Sachs і Morgan Stanley змінили свою вивіску, перестали бути інвестиційними банками в зв'язку з особливими ризиками і необхідністю отримати додаткову підтримку Федеральної резервної системи. Це стало причиною кризи довіри між банками і різкого зростання ставок кредитування, що особливо сильно позначилося на кредитних ринках що розвиваються.

Країни, економіки яких розвиваються, вали в тяжку кризу аж до 2013 року, а деякі не вийшли з неї до сих пір. Україна являється однією з найбільш постраждалих країн від даної кризи. За період 2008-2009 рік ВВП України впав на 14.9%, а гривня девальвувала в 2 рази.

Приклад цієї кризи показує наскільки глобальним став світ, і як ризикові махінації банків в одній країні можуть призвести до світової кризи.

1.8 Постулати технічного аналізу

Перш ніж приступати до вивчення методів і засобів, що використовуються для технічного аналізу фінансових ринків, необхідно перш за все визначити, що ж, власне, являє собою технічний аналіз. Крім того, слід зупинитися на його філософській основі, провести чітке розмежування між технічним і фундаментальним аналізом і, нарешті, згадати про критичні зауваження, яким нерідко піддається технічний аналіз.

Отже, приступимо до визначення. Технічний аналіз - це дослідження динаміки ринку, найчастіше за допомогою графіків, з метою прогнозування майбутнього напрямку руху цін. Термін "динаміка ринку" включає в себе три основних джерела інформації, що знаходяться в розпорядженні технічного аналітика, а саме: ціну, обсяг і відкритий інтерес.

Отже, сформулюємо три постулати, на яких, як на трьох китах, стоїть технічний аналіз:

1. Ринок враховує все.
2. Рух цін підпорядкований тенденціям.
3. Історія повторюється.

1.8.1 Ринок враховує все

Дане твердження, по суті справи, є основним правилом всього технічного аналізу.

Технічний аналітик вважає, що причини, які хоч якось можуть вплинути на ринкову вартість цінного паперу (а причини ці можуть бути найрізноманітнішого походження: економічні, політичні, психологічні - будь-які), неодмінно знайдуть

своє відображення в ціні на цей товар. Із цього слідує, що все, що від аналітика вимагається, - це ретельне вивчення руху цін.

Отже, іншими словами, будь-які зміни в динаміці попиту і пропозиції відображаються на русі цін. Якщо попит перевищує пропозицію, ціни зростають. Якщо пропозиція перевищує попит, ціни йдуть вниз. Це, власне, лежить в основі будь-якого економічного прогнозування. А технічний аналітик підходить до проблеми з іншого кінця і міркує таким чином: якщо з яких би то не було причин ціни на ринку пішли вгору, значить, попит перевищує пропозицію. Отже, по макроекономічними показниками ринок вигідний для биків. Якщо ціни падають, ринок вигідний для ведмедів.

Багато фахівців з технічного аналізу погодяться, що саме глибинні механізми попиту та пропозиції, економічна природа того чи іншого товарного ринку і визначають динаміку на підвищення або на зниження цін. Самі по собі графіки не роблять на ринок ані найменшого впливу, вони лише відображають психологічну тенденцію на підвищення або на пониження, яка в даний момент оволодіває ринком.

Як правило, фахівці з аналізу графіків вважають за краще не вдаватися в глибинні причини, які зумовили зростання або падіння цін. Дуже часто на ранніх стадіях, коли тенденція до зміни цін тільки-тільки намітилася, або, навпаки, в якісь переломні моменти причини таких змін можуть бути нікому не відомі. Може здатися, що технічний підхід надмірно спрощує і обмежує завдання, але логіка, яка стоїть за першим вихідним постулатом - "ринок враховує все", - стає тим очевидніше, чим більше досвіду реальної роботи на ринку набуває технічний аналітик.

З цього випливає, що все те, що будь-яким чином впливає на ринкову ціну, неодмінно на цій самій ціні відіб'ється. Отже, необхідно лише уважно відстежувати і вивчати динаміку цін. Аналізуючи цінові графіки і безліч

додаткових індикаторів, технічний аналітик добивається того, що ринок сам вказує йому найбільш ймовірний напрямок свого розвитку. Нам немає потреби намагатися переіграти або перехитрити ринок. Технічний аналітик знає, що з якихось причин ринок йде вгору або вниз. Але навряд чи знання того, що ж це за причини, необхідно для його прогнозів.

1.8.2 Рух цін підпорядкований тенденціям

Поняття тенденції або тренда - одне з основоположних понять в технічному аналізі. Необхідно засвоїти, що, по суті справи, все, що відбувається на ринку, підпорядковане тим чи іншим тенденціям. Основна мета складання графіків динаміки цін полягає в тому, щоб виявляти ці тенденції на ранніх стадіях їх розвитку і торгувати відповідно до їх зміни. Більшість методів технічного аналізу опираються на тенденції, тобто їх функція полягає в тому, щоб допомогти аналітику розпізнати тенденцію і слідувати їй протягом усього періоду її існування.

З положення про те, що рух цін підпорядкований тенденціям, виникають два висновки:

- діюча тенденція, цілком ймовірно, буде розвиватися далі, а не перетворюватись у власну протилежність.
- діюча тенденція буде розвиватися доти, поки не почнеться рух у зворотному напрямку.

Який би словесної параболою не здавалася це положення, слід твердо запам'ятати, що всі методи прямування тенденцій ґрунтуються на тому, що торгівля в напрямку існуючої тенденції повинна вестися до тих пір, поки тенденція не подасть ознак розвороту.

1.8.3 Історія повторюється

Технічний аналіз і дослідження динаміки ринку найтіснішим чином пов'язані з вивченням людської психології. Наприклад, графічні цінові моделі, які були виділені і класифіковані протягом останніх ста років, відображають важливі особливості психологічного стану ринку. Перш за все вони вказують, які очікування— росту чи падіння— панують в даний момент на ринку. І якщо в минулому ці моделі працювали, є всі підстави припускати, що і в майбутньому вони будуть працювати, бо ґрунтуються вони на людській психології, яка з роками не змінюється. Можна сформулювати постулат - "історія повторюється" - дещо іншими словами: ключ до розуміння майбутнього криється у вивченні минулого. А можна і зовсім інакше: майбутнє - це всього лише повторення минулого.

1.9 В чому різниця між Технічним і Фундаментальним аналізом?

Якщо технічний аналіз в основному займається вивченням динаміки ринку, то предметом досліджень фундаментального аналізу є економічні сили попиту і пропозиції, які викликають коливання цін, тобто змушують їх іти вгору, вниз або зберігатися на існуючому рівні. При фундаментальному підході аналізуються всі чинники, які так чи інакше впливають на ціну товару. Робиться це для того, щоб визначити внутрішню або дійсну вартість товару. Згідно з результатами фундаментального аналізу, саме ця дійсна вартість і відображає, скільки насправді вартий той чи інший товар. Якщо дійсна вартість нижче ринкової ціни товару, значить товар потрібно продавати, так як за нього дають більше, ніж він коштує на справі. Якщо дійсна вартість вище ринкової ціни товару, значить

потрібно купувати, бо він йде дешевше, ніж коштує насправді. При цьому виходять виключно із законів попиту і пропозиції.

Обидва ці підходи до прогнозування динаміки ринку намагаються вирішити одну і ту ж проблему, а саме: визначити, в якому ж напрямку будуть рухатися ціни. Але до проблеми цієї вони підходять з різних кінців. Якщо фундаментальний аналітик намагається розібратися в причині руху ринку, технічного аналітика цікавить тільки факт цього руху. Все, що йому потрібно знати, це те, що такий рух або динаміка ринку має місце, а що саме її викликало - не так уже й важливо. Фундаментальний аналітик буде намагатися з'ясувати, чому це сталося.

Багато фахівців, що працюють з акціями, традиційно відносять себе або до технічних, або до фундаментальних аналітиків. Насправді ж межа тут дуже розмита. Багато фундаментальних аналітиків мають хоча б початкові навички аналізу графіків. У той же час, немає такого технічного аналітика, який хоча б у загальних рисах не уявляв собі основних положень фундаментального аналізу. Справа в тому, що дуже часто ці два методи аналізу дійсно вступають в протиріччя один з одним. Зазвичай в самому зародку якихось важливих зрушень, поведінка ринку не вписується в рамки фундаментального аналізу і не може бути пояснена на основі тільки економічних факторів. Саме в ці моменти, моменти для загальної тенденції найкритичніші, два види аналізу - технічний і фундаментальний - і розходяться найбільше. Пізніше, на якомусь етапі вони співпадають по фазі, але, як правило, занадто пізно для адекватних дій трейдера.

Одним з пояснень цього уявного протиріччя є наступне: ринкова ціна випереджає всі відомі фундаментальні дані. Іншими словами, ринкова ціна служить випереджальним індикатором фундаментальних даних або міркувань здорового сенсу. У той час як ринок вже врахував всі відомі економічні фактори, ціни починають реагувати на якісь зовсім нові, ще не відомі фактори.

Найзначніші періоди зростання і падіння цін в історії починалися в обстановці, коли нічого або майже нічого, з точки зору фундаментальних показників, не віщувало ніяких змін. Коли ж ці зміни ставали зрозумілі фундаментальним аналітикам, нова тенденція вже розвивалася в повну силу.

Згодом до технічного аналітику приходить впевненість у своєму вмінні читати і аналізувати графіки. Він поступово звикає до ситуації, коли динаміка ринку не збігається з «здоровим глуздом». Йому починає подобатися бути в меншості.

Технічний аналітик твердо знає, що рано чи пізно причини динаміки ринку стануть всім відомі. Але це буде потім. А зараз не можна витратити час на очікування цього додаткового підтвердження власної правоти.

Отже, можна зрозуміти, у чому заключається перевага технічного аналізу над фундаментальним. Якщо необхідно вибрати якийсь один з двох підходів, то, логічніше, це, повинен бути технічний аналіз. По-перше, за визначенням, він включає в себе дані, якими оперує фундаментальний аналіз, адже якщо вони відображаються на ринковій ціні, значить їх уже немає необхідності аналізувати окремо. Так що аналіз графіків, по суті справи, стає спрощеною формою фундаментального аналізу. Між іншим, цього не можна сказати про останній. Фундаментальний аналіз не займається вивченням динаміки цін. Можна успішно працювати на фондовому ринку, користуючись тільки технічним аналізом. Але дуже мало ймовірно, що аналітик досягне хоч скільки-небудь успіху, якщо буде спиратися тільки на дані фундаментального аналізу.

Однією з сильних сторін технічного аналізу безсумнівно, є те, що його можна використовувати практично для будь-якого засобу торгівлі і в будь-якому часовому інтервалі. Немає такої області в операціях на фондовій і на товарній біржі, де б не застосовувалися методи технічного аналізу.

Якщо мова йде про товарні ринки, то технічний аналітик, завдяки своїм

графіками, може відстежувати ситуацію на будь-якій кількості ринків, чого не можна сказати про фундаменталістів. Останні, як правило, використовують таку кількість різних даних для своїх прогнозів, що просто змушені спеціалізуватися на якомусь одному ринку або групі ринків, переваги ж широкої спеціалізації очевидні.

Перш за все, у будь-якого ринку існують періоди сплеску активності і періоди летаргії, періоди яскраво вираженої тенденції цін і періоди невизначеності. Технічний аналітик вільно може сконцентрувати всю свою увагу і сили на тих ринках, де тенденції цін виразно простежуються, а всіма іншими поки знехтувати. Іншими словами, він максимально використовує переваги мінливого ринку, а на практиці це виражається в переключенні уваги і засобів. У різні періоди часу ті чи інші ринки раптом починають "вирувати", ціни на них утворюють чіткі тенденції, а потім активність згасає, ринок стає млявим, динаміка цін - невизначеною. Але на якомусь іншому ринку в цей момент раптом починається спалах активності. І технічний аналітик в подібній ситуації має свободу вибору, чого не можна сказати про фундаменталістів, вузька спеціалізація яких на якомусь певному ринку або групі ринків просто-напросто позбавляє їх цієї можливості маневру. Навіть якщо фундаментальний аналітик і вирішить переключитися на що-небудь інше, маневр цей зажадає від нього набагато більше часу і зусиль.

Ще однією сильною стороною технічного аналізу є можливість його застосування на будь-якому відрізку часу. І зовсім не важливо, чи граєте ви на коливаннях в межах одного дня торгів, коли важливий кожен крок, або аналізуєте середньострокову тенденцію, в будь-якому випадку ви використовуєте одні і ті ж принципи. Іноді висловлюється думка, що технічний аналіз ефективний лише для короткострокового прогнозування. Насправді ж це не так. Дехто помилково вважає, що для довгострокових прогнозів більш підходить фундаментальний

аналіз, а доля технічних факторів - лише короткостроковий аналіз з метою визначення моментів входу і виходу з ринку. Але, як показує практика, використання тижневих і місячних графіків, охоплює динаміку ринку за кілька років, що для довгострокового прогнозування виявляється надзвичайно плідним. Важливо до кінця зрозуміти основні принципи технічного аналізу, щоб відчувати ту гнучкість і свободу маневру, яку вони надають аналітику, дозволяючи йому з рівним успіхом застосовувати їх для аналізу будь-якого засобу торгівлі і на будь-якому відрізку часу.

1.10 Висновки до розділу

Отже, в цьому розділі було розглянуто основні принципи та етапи «життя» фондового ринку. Ми розглянули, що таке біржа та як вона працює; як за допомогою цифр можна зрозуміти що відбувається всередині ринку; як всесвітня економіка реагує на вартість цінних паперів; яв сьогодні повторюється те, що було 100 років тому.

Таким чином, можна зрозуміти, чому, здавалось би, вторинний ринок з торгівлі цінними паперами так сильно впливає на усю світову економіку.

Тож щоб зможти управляти своїми активами, або прогнозувати розвиток цього ринку, потрібно навчитись його аналізувати, що буде розглянуто в наступному розділі.

РОЗДІЛ 2 МАТЕМАТИЧНІ МЕТОДИ ТА НАПРЯМКИ ПРОГНОЗУВАННЯ ДИНАМІКИ ФОНДОВИХ РИНКІВ

2.1 Математичні моделі часових рядів

Часовий ряд - ряд спостережуваних значень досліджуваної величини, упорядкований в часі. Часовий ряд істотно відрізняється від вибірки: значення розподілені неоднаково і вони не є незалежними, тобто не можна розташувати досліджувані значення в будь-якому порядку.

Існують дві основні мети аналізу часових рядів: визначення природи ряду і прогнозування - передбачення майбутніх значень часового ряду по теперішнім і минулим значенням. Обидві ці цілі вимагають, щоб модель ряду була ідентифікована і формально описана. Як тільки модель визначена, можна з її допомогою інтерпретувати отримані дані.

Аналіз часових рядів припускає, що дані містять систематичну складову (зазвичай включає кілька компонент) і випадковий шум (помилку).

Більшість регулярних складових часових рядів належить до двох класів: вони є або трендом, або сезонною складовою.

- Тренд є загальною систематичною лінійною або нелінійною компонентою, яка може змінюватися в часі.
- Сезонна складова - це компонента, що періодично повторюється. Сезонна залежність може бути формально визначена як кореляційна залежність порядку k між кожним i -м елементом ряду і $(i-k)$ -м елементом, k зазвичай називають лагом.

Обидва ці виду регулярних компонент часто присутні в ряді одночасно. Наприклад тренд показує що виручка компанії постійно росте, а в такі дні як

Різдво, Новий Рік або ж Чорна п'ятниця виручка збільшується в декілька разів ніж в інші дні, це сезонна складова цього ряду.

Білий шум - стаціонарний ряд, що представляє собою набір випадкових величин, незалежних одна від одної з нульовим математичним очікуванням і постійною дисперсією.

У більшості випадків часовий ряд можна представити як суму або добуток трендової, сезонної і випадкової компонент. Модель, в якій часовий ряд представлений як сума компонент, називається адитивною моделлю часового ряду. Модель, в якій тимчасової ряд представлений як добуток компонент, називається мультиплікативної моделлю часового ряду.

Не існує «автоматичного» способу виявлення тренду в часовому ряду. Якщо тимчасові ряди містять значну помилку, то першим кроком виділення тренду є згладжування.

Згладжування завжди включає деякий спосіб локального усереднення даних, при якому несистематичні компоненти взаємно погашають один одного. Основний метод згладжування - ковзне середнє, в якому кожен член ряду замінюється простим або зваженим середнім n сусідніх членів, де n - ширина «вікна». Замість середнього можна використовувати медіану значень, що потрапили у вікно. Основна перевага медіанного згладжування, в порівнянні зі згладжуванням ковзним середнім, полягає в тому, що результати стають більш стійкими до викидів (наявними всередині вікна). Таким чином, якщо в даних є викиди (пов'язані, наприклад, з помилками вимірювань), то згладжування медіаною зазвичай призводить до більш гладких або, принаймні, більш «надійних» кривих, в порівнянні з ковзним середнім з тим же самим вікном. Основний недолік медіанного згладжування в тому, що при відсутності явних викидів, він призводить до більш «зубчастих» кривих (чим згладжування ковзним середнім) і не дозволяє використовувати ваги.

Багато монотонних часових рядів можна добре наблизити лінійною функцією. Якщо ж є явна монотонна нелінійна компонента, то дані спочатку слід перетворити, щоб усунути нелінійність. Зазвичай для цього використовують логарифмічне, експоненціальне або (менш часто) поліноміальне перетворення даних.

2.2 Модель Бокса-Дженкінса ARIMA

В реальних даних дуже часто немає чітко вираженої регулярної складової, а помилки в окремих спостереженнях іноді бувають дуже значущими. Тому для того щоб виділити регулярні компоненти, а також побудувати прогноз часто використовують спеціально розроблені методи.

ARIMA (autoregressive integrated moving average)- методологія розроблена Боксом та Дженкінсом в 1976 дозволяє виділяти усі компоненти та прогнозувати динаміку ряду. Даний метод надзвичайно популярний і використовується на практиці в більшості додатків та програм. Але ARIMA досить складний метод, та потребує великої кваліфікації від користувача. Розглянемо основні ідеї даного методу, та методи, що лежать в основі методології.[12,13]

– Процес авторегресії, AR(p)

Більшість часових рядів містять елементи, які послідовно залежать один від одного. Таку залежність можна виразити наступним рівнянням:

$$x_t = \xi + \phi_1(x_{t-1}) + \phi_2(x_{t-2}) + \phi_3(x_{t-3}) + \dots + \varepsilon, \quad (2.1)$$

де, ξ – вільний член, а $\bar{\phi}_{1,n}$ – параметри авторегресії.

Кожне спостереження це сума випадкової компоненти ε та лінійної комбінації минулих спостережень. Слід зазначити, що процес авторегресії буде

стаціонарним, якщо його параметри лежать в певному діапазоні, в протилежному випадку, попередні значення будуть накопичуватись і значення наступних x_t можуть бути необмеженими, відповідно, ряд буде не стаціонарним.

– Процес ковзного середнього, MA(q)

В процесі ковзного середнього на кожний елемент ряду впливає сумарна дія минулих помилок, це можна виразити такою формулою:

$$x_t = \mu + \varepsilon_t - \theta_1 \varepsilon_{t-1} - \theta_2 \varepsilon_{t-2} - \theta_3 \varepsilon_{t-3} - \dots, \quad (2.2)$$

де, μ це константа, а $\bar{\theta}_{1,n}$ – параметри ковзного середнього. Тобто, поточне спостереження ряду є сумою випадкової компоненти та лінійної комбінації випадкових компонент на минулих моментах часу.

Варто зазначити, що існує двоїстість між процесами ковзного середнього і авторегресії. Це означає, що наведене вище рівняння змінного середнього можна переписати у вигляді рівняння авторегресії (необмеженого порядку), і навпаки. Це так зване властивість оборотності.

– Процес авторегресії ковзного середнього, ARMA(p,q)

Моделлю ARMA(p,q), де p і q - цілі числа, що задають порядок моделі, називається наступний процес генерації часового ряду x_t :

$$x_t = c + \varepsilon_t + \sum_{i=1}^p \alpha_i x_{t-i} + \sum_{i=1}^q \beta_i \varepsilon_{t-i}, \quad (2.3)$$

де, c - константа, ε_t – білий шум, а α_i та β_i це авторегресійні коефіцієнти та коефіцієнти ковзного середнього відповідно.

Отже, ARMA(p,q) це лінійна модель множинної регресії, в якій в якості пояснювальних змінних виступають минулі значення самої залежної змінної, а в якості регресійного залишку – ковзне середнє з елементів білого шуму. ARMA

процеси мають більш складну структуру в порівнянні зі схожими AR або MA процесами в чистому вигляді, але при цьому ARMA процеси характеризуються меншою кількістю параметрів, що перевагою даної моделі.

Для побудови моделі ARMA за допомогою серії спостережень необхідно визначити порядок моделі (числа p і q), а потім і самі коефіцієнти. Для визначення порядку моделі може застосовуватися дослідження таких характеристик часового ряду, як його автокореляційна функція і часткова автокореляційна функція. Для визначення коефіцієнтів застосовуються такі методи, як метод найменших квадратів і метод максимальної правдоподібності.

Але дана модель може використовуватись лише для стаціонарних рядів. Тому для нестаціонарних рядів використовують ARIMA моделі.

– Інтегрована модель авторегресії ковзного середнього, ARIMA(p, d, q)

ARIMA -модель Бокса – Дженкінса є розширенням моделі ARMA для нестаціонарних часових рядів, які можна зробити стаціонарними взявши різниці деякого порядку від вихідного часового ряду. Модель ARIMA(p, d, q) означає, що різниці часового ряду порядку d підкоряються моделі ARMA(p, q).

Модель можна записати в вигляді:

$$\Delta^d X_t = c + \varepsilon_t + \sum_{i=1}^p \alpha_i \Delta^d X_{t-i} + \sum_{j=1}^q \beta_j \varepsilon_{t-j}, \quad (2.4)$$

де ε_t це стаціонарний часовий ряд, c, α_i, β_j – параметри моделі, а Δ^d - оператор різниці часового ряду порядку d . По іншому дану модель можна представити як ARMA($p+d, q$).

Побудову моделі Бокса-Дженкінса зазвичай виконують в 3 етапи, розглянемо їх нижче.[11]

- 1) Ідентифікація параметрів моделі. Для того щоб побудувати модель, необхідно, в першу чергу підібрати вхідні параметри. Щоб привести

ряд до стаціонарності беруть різницю часового ряду, що визначається параметром d . Основними інструментами знаходження порядку є графіки, автокореляційна функція (АКФ), та часткова автокореляційна функція (ЧАКФ). Варто зазначити, що надмірна кількість взятих різниць призводить до менш стабільних коефіцієнтів. Часто для перевірки стаціонарності використовують тест Дікі-Фулера. Також на цьому кроці також визначають які параметри авторегресії (p) та ковзного середнього (q) варто вибрати.

- 2) Оцінка параметрів моделі, де використовуються процедури мінімізації функції втрат.
- 3) Оцінка самої моделі відбувається за допомогою прогнозу ряду на інтервалі, з уже відомими даними, а потім порівняння прогнозу з цими ж даними. Варто зазначити що якісна модель повинна не лише давати досить гарний прогноз, а ще й бути економною, тобто мати найменшу кількість параметрів, та найбільшу кількість ступенів свободи серед усіх підігнаних моделей.

Хоча моделі, побудовані за допомогою часових рядів використовуються в великій кількості професійних додатків, створених для прогнозування динаміки фондових ринків, зараз, їх усе частіше доповнюють, або навіть замінюють моделями, що створені на базі штучних нейронних мереж. Звичайно, відомо що усі макроекономічні показники ринку цінних паперів підпорядковуються певним циклам, наприклад, цикли Кітчина (2-4 роки), цикли Жюгляра (7-11 років), або цикли Кондратьєва (48-55 років). Але акції окремої компанії зазвичай не підпорядковані цим тенденціям, адже на них впливають політичні, соціально-економічні або соціальні фактори, які з точки зору часових рядів є випадкові та хаотичні. Тому, фактично, прогноз хаотичного часового ряду є дуже складною задачею.

2.3 Методи нейронних мереж

Сьогодні існує безліч технологій, що використовують штучні нейронні мережі. Причому спектр задач, що виконують такі технології дуже і дуже широкий, починаючи з розпізнавання обличч та рукописного тексту, закінчуючи ботами в онлайн іграх та чат-ботами в соціальних мережах. Для нашої задачі прогнозування варто зосередити увагу на рекурентних нейронних мережах.

RNN (Recurrent neural network), Рекурентна нейронна мережа починає свій розвиток з мережі Хопфілда. У 1993 нейронна система запам'ятовування і стиснення історичних даних змогла вирішити задачу «дуже глибокого навчання», в якій в рекурентній мережі розгорталось більше 1000 послідовних шарів. Дана технологія базується на використанні минулих станів мережі, для визначення теперішнього. Такі мережі на практиці зазвичай використовують, коли вхідні данні задачі представляють собою нефіксовану послідовність значень. На кожному кроці «навчання» t значення $h_t \in \mathbb{R}^m$ прихованого шару рекурентної нейронної мережі визначається наступним чином:

$$h_t = f(Wx_t + Uh_{t-1} + b^h), \quad (2.5)$$

де, $x_t \in \mathbb{R}^m$ – вхідний вектор в момент часу t , $W \in \mathbb{R}^{m \times n}$, $U \in \mathbb{R}^{m \times m}$, $b^h \in \mathbb{R}^m$ – параметри рекурентної нейронної мережі, що навчаються, f – функція нелінійного перетворення. Найчастіше в якості нелінійного перетворення використовують одну з наступних функцій:

- Сигмоїда: $f(x) = \frac{1}{1+e^{-x}}$
- Гіперболічний тангенс: $f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$
- Функція-випрямляч ReLU: $f(x) = \max(0, x)$

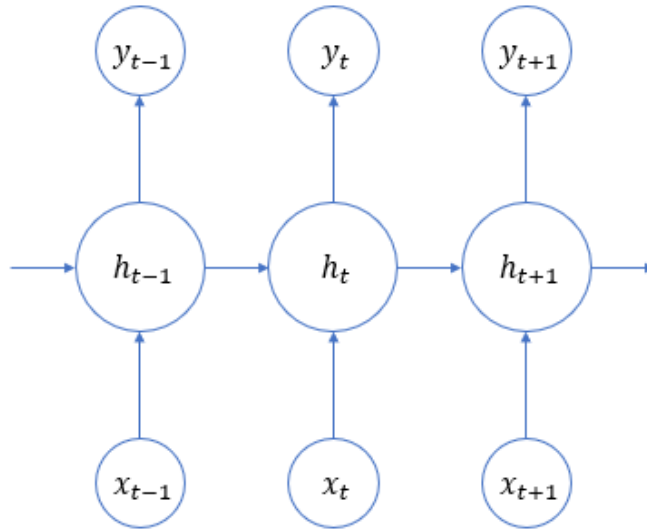


Рис. 2.1 — Проста рекурентна нейронна мережа

В простій рекурентній нейронній мережі, що показана на рисунку 2.1, вихідне значення y_t на кроці t рахується за формулою:

$$y_t = Wh_t + b, \quad (2.6)$$

де W , b -навчальні параметри.

Проста РНМ дуже гарно працює для задач підбору контексту на базі попередніх кроків збереження інформації. Наприклад підбір фрагментів для відеоролику, або передбачення наступного слова в тексті. Але існують випадки коли потрібно звернутись до інформації, що була згадана деяку кількість кроків назад. І на жаль, на практиці, при збільшенні розриву РНМ втрачають зв'язок між інформацією. Тому, щоб вирішити дану проблему, був розроблений метод LSTM нейронних мереж.

2.4 LSTM нейронні мережі

В 1997 році Зеппом Хохрайтером та Юргеном Шмідгубером був розроблений метод LSTM (Long Short-Term Memory) - метод Довгої короткочасної пам'яті для рекурентних нейронних мереж. Основна ідея даного методу це виділення окремої комірки пам'яті, для зберігання інформації, отриманої в минулих моментах часу. Для контролю цієї комірки використовуються три «шлюзи» або «вентилі»:

- Вентиль пам'яті f_t відповідає за те, щоб модель під час тренування сформувала спеціальний механізм забування: коли приходить нова вхідна інформація, модель повинна знати, які знання слід продовжувати пам'ятати, а які слід забути.
- Вентиль додавання або вхідний вентиль i_t визначає які елементи вхідної інформації можуть бути корисними, та додає їх в довгострокову пам'ять.
- Вентиль фокусування або вихідний o_t визначає яку інформацію можна використати в даний момент, а яка можливо, буде корисною в наступних кроках. Тобто мережа вчиться фокусуватись на окремих деталях інформації.

Отже, нижче описано алгоритм, за яким працюватиме дана нейрона мережа, схема роботи така ж як і для РНМ, та демонструється на Рис. 1.

В момент часу t мережа отримує нові вхідні дані x_t . Довгострокова та робоча пам'ять, що представлені у вигляді n -мірних векторів, надходять з попереднього моменту часу, і обидва види пам'яті мають бути оновлені.

Почнемо з довгострокової пам'яті, по-перше, мережа визначає які елементи потрібно забути, а які продовжувати пам'ятати. Вона використовує робочу пам'ять та вхідні дані, для того щоб вирахувати вентиль пам'яті f_t . Цей вентиль являється n -мірним вектором з значеннями від 0 до 1, кожне з яких визначає, яку

частину даного елементу довгострокової пам'яті необхідно продовжувати пам'ятати. Наприклад, 1 означає продовжувати пам'ятати елемент, а 0 означає повністю його забути. Далі необхідно визначити інформацію, яку ми можемо витягти з x_t , і яка буде кандидатом на додавання в довгострокову пам'ять. Але перш ніж додавати інформацію, необхідно визначити, які її елементи дійсно можуть бути корисні, для цього розраховують вентиль входу i_t .

Наступним кроком, необхідно оновити робочу пам'ять. Модель повинна навчитись виділяти зі своєї довгострокової пам'яті інформацію, яка може бути корисною в найближчий час, тобто мережа має фокусувати свою увагу, для цього розраховують вихідний вентиль o_t .

Усе ці кроки можна представити у вигляді наступних формул:

$$f_t = \sigma_g(W_f x_t + U_f h_{t-1} + b_f) \quad (2.7)$$

$$i_t = \sigma_g(W_i x_t + U_i h_{t-1} + b_i) \quad (2.8)$$

$$o_t = \sigma_g(W_o x_t + U_o h_{t-1} + b_o) \quad (2.9)$$

$$h_t = f_t \circ c_{t-1} + f_t \circ \sigma_c(W_c x_t + U_c h_{t-1} + b_c) \quad (2.10)$$

$$y_t = o_t \circ \sigma_c(h_t), \quad (2.11)$$

де:

x_t – вектор вхідних даних;

y_t – вектор вихідних даних;

h_t – вектор стану комірки пам'яті;

W , U та b – матриці та вектор параметрів;

f_t , i_t , o_t – вектори вентилів;

σ – це функція активації, σ_g -сигмоїдна функція, σ_c – функція гіперболічного тангенсу.

На практиці існує дуже багато варіацій алгоритму LSTM, але усіх їх об'єднує більш досконале використання пам'яті ніж в інших РНМ моделях. LSTM мережі навчаються за допомогою алгоритму Зворотного поширення в часі (backpropagation through time), ідея якого складається з розгортання графа обчислень в часі. Даний алгоритм є значно швидшим для тренування рекурентних нейронних мереж, ніж методики оптимізації загального призначення, такі як еволюційна оптимізація. [14]

2.5 Методологія програмного пакету Prophet

В 2017 році представники компанії Facebook Шон Тейлор та Бенджамін Летхем представили розроблений пакет для прогнозування часових рядів – prophet («пророк»). Даний пакет дозволяє створювати точні моделі в автоматичному режимі. Prophet доступний для вільного скачування, та представляє собою бібліотеку для Python та R. Розглянемо методологію, що використовується в даному пакеті.

В основі методології лежить процедура підгонки адитивних регресійних моделей (Generalized Additive Models, GAM) наступного вигляду:

$$y(t) = g(t) + s(t) + h(t) + \epsilon_t, \quad (2.12)$$

де, $g(t)$ – тренд функція, $s(t)$ – сезонні коливання, $h(t)$ – функція, що відображає ефекти святкових днів та інших впливових подій, ϵ_t – нормально розподілені випадкові збурення. Розглянемо ці елементи та методи їх апроксимацій більш детально:

1) Тренд моделюється за допомогою кусково-лінійної регресії або логістичної кривої росту. Стандартна логістична функція для опису росту в системі з обмеженою ємкістю має вигляд:

$$g(t) = \frac{C}{1 + \exp(-k(t-m))}, \quad (2.13)$$

де C - ємкість системи, k - швидкість росту, t - час, а m - параметр, що рухає функцію вздовж осі часу. Однак дана модель не враховує декілька факторів. По-перше ємкість багатьох систем не постійна, наприклад кількість людей що мають доступ до Інтернету, з часом росте. Тому в пакеті prophet ємкість системи C замінена на динамічну $C(t)$.

По-друге, не постійною зазвичай буває і швидкість росту k . Для цього в даній моделі водиться поняття точок зламу тренду. Припустимо, що S таких точок знаходяться на часових мітках $s_j, j = 1, 2, \dots, S$. Сукупність усіх змін швидкості росту δ_j можна представити у вигляді вектора $\delta \in \mathbb{R}^S$. Тоді швидкість росту в будь якій точці часу t буде рівнятись сумі базової швидкості k та всіх змін, що передували цій точці: $k + \sum_{j:t > s_j} \delta_j$. Більш наочно це можна представити за допомогою такого вектора як $a(t) \in \{0,1\}^S$, або:

$$a_j(t) = \begin{cases} 1, & \text{if } t \geq s_j \\ 0, & \text{else} \end{cases} \quad (2.14)$$

Тоді швидкість росту в момент часу t складе $k + a(t)^T \delta$. При зміні швидкості росту необхідно також змінити параметр m , щоб забезпечити гладке з'єднання сегментів кривої тренда на відповідному часовому відрізку. Така поправка:

$$\gamma_j = (s_j - m - \sum_{l < j} \gamma_l) \left(1 - \frac{k + \sum_{l < j} \delta_l}{k + \sum_{l \leq j} \delta_l}\right) \quad (2.15)$$

Отже, кусково-логістична функція, що використовується в пакеті prophet має вигляд:

$$g(t) = \frac{c(t)}{1 + \exp(-(k + a(t)^T \delta)(t - (m + a(t)^T \gamma)))} \quad (2.16)$$

При необмеженому ж рості системи замість представленої вище функції, в пакеті prophet використовується кусково-лінійна функція наступного виду:

$$g(t) = (k + a(t)^T \delta)t + (m + a(t)^T \gamma) \quad (2.17)$$

2) Річна та тижнева сезонність моделюється за допомогою рядів Фур'є, виду:

$$s(t) = \sum_{n=1}^N \left(a_n \cos\left(\frac{2\pi n t}{P}\right) + b_n \sin\left(\frac{2\pi n t}{P}\right) \right), \quad (2.18)$$

де, для річної сезонності $P=365.25$, $N=10$, а для тижневої – $P=7$, $N=3$. Параметр N визначає гладкість функції.

3) "Свята" (наприклад, офіційні святкові та вихідні дні - Новий рік, Різдво і т.д., а також інші дні, під час яких властивості часового ряду можуть істотно змінитися - спортивні або культурні події, природні явища, тощо.): представлені у вигляді індикаторних змінних.

Можливість швидко і зручно створювати хороші моделі є основною перевагою Prophet. [15]

РОЗДІЛ 3 ПОРІВНЯЛЬНИЙ АНАЛІЗ РЕЗУЛЬТАТІВ МОДЕЛЮВАННЯ ТА ПРОГНОЗУВАННЯ ДИНАМІКИ ФОНДОВОГО РИНКУ

В цьому розділі усі дослідження були проведенні за допомогою мови програмування Python. Дані та графіки були візуалізовані в середовищі розробки Jupyter Notebook.

3.1 Прогнозування

Зазвичай, основною ціллю побудови моделей часового ряду є отримання прогнозних значень в деякий майбутній момент часу.

Припустимо, що в момент часу t необхідно отримати прогноз Y_{t+h} , тобто значення змінної Y через h тактів часу. На момент прогнозування при моделюванні часового ряду інформаційна множина I_t , на яку спирається прогноз, містить значення змінної Y_t і усіх її лагів. В загальному випадку, прогноз $Y_{t+h|t}$ являється функцією змінних цієї множини.

Для вибору оптимальної моделі необхідно мінімізувати математичне очікування квадрату помилки прогнозу:

$$E\{(Y_{t+h} - \hat{Y}_{t+h|t})^2 | I_t\} \rightarrow \min \quad (3.1)$$

Виходячи із цього, можна показати, що найкращим прогнозом є умовне математичне очікування Y_{t+h} при заданій інформації I_t :

$$Y_{t+h|t} = E\{Y_{t+h} | I_t\} \quad (3.2)$$

Таким чином, оптимальний прогноз задовольняє всім звичайним властивостями операторів математичного очікування. В тому числі, умовне математичне сподівання суми це сума умовних математичних сподівань.

3.2 Методи оцінки параметрів моделі

Якщо ми розглянемо нормально розподілену сукупність спостережень із середнім значенням μ та стандартним відхиленням σ , то ми маємо $x_1, \dots, x_n \sim N(\mu, \sigma^2)$ і середнє значення вибірки $\bar{x} = \frac{x_1 + \dots + x_n}{n}$ є випадковою змінною, розподіленою таким чином, що $\bar{x} \sim N(\mu, \frac{\sigma^2}{n})$. Тоді статистичні похибки є $e_i = x_i - \mu$, а залишки це $r_i = x_i - \bar{x}$.

Перечислимо основні методи, за допомогою яких можна оцінити якість побудованої моделі:

- Середньоквадратичне відхилення (STD) розраховується за формулою:

$$\sigma = \sqrt{\sigma^2} = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - \bar{x})^2}, \quad (3.3)$$

де $\vec{x}_i = \overrightarrow{1, N}$ – значення спостережень даної вибірки, а \bar{x} середнє значення цих спостережень.

- Середньоквадратична похибка (MSE) розраховується як σ^2 .
- Середня абсолютна похибка вимірювання (MAE) розраховується як:

$$MAE = \frac{1}{N} \sum_{i=1}^N |x_i - \bar{x}| \quad (3.4)$$

– Середня абсолютна відсоткова похибка:

$$MAPE = \frac{1}{N} \sum_{i=1}^N \frac{|x_i - \bar{x}|}{x_i} \quad (3.5)$$

Цей коефіцієнт розраховують в відсотках, і він фактично вказує який відсоток від значень складає помилка.

3.3 Вхідні дані

Дані, з якими ми будемо працювати в цьому розділі представляють собою регулярні спостереження за ціною однієї акції конкретної компанії. Таку інформацію досить легко знайти в інтернеті. В цій роботі дані були взяті з таких ресурсів як: <https://finance.yahoo.com/> - дає можливість легко та швидко скачати масиви інформації про ціни акцій, індексів, товарів за конкретний вказаний період та з заданою частотою; бібліотека для Python Quandl має щоденно поновлювальну базу з цінами акцій 3000 американських компаній.

Акції яких саме компаній варто розглянути в цьому дослідженні, а яких ні? По-перше, не варто брати ті компанії, на ціни акцій яких, мають великий вплив соціальні на інформаційні чинники. Можна привести в приклад компанію Tesla, що фактично являється стартапом. Думки спеціалістів розмежовуються на два табори: перші вважають що ця компанія є майбутнім бізнесу, а інші – що її вартість лише велика фінансова бульбашка. Тому ціни акцій цієї компанії знаходяться в постійному коливанні.

Розглянемо дані для компанії Apple в проміжку 01.01.1995 – 24.05.2020 з частотою спостережень в один тиждень. Перші 5 елементів наведено на рис. 3.1. Ця компанія має досить міцну репутацію та динаміка її ціни має чіткий тренд.

Вибірка включає 1326 спостережень. З усієї таблиці нас цікавитимуть тільки стовбці “Date” та “Close”, за якими ми можемо побудувати графік (рис. 3.2).

Date	Open	High	Low	Close	Adj Close	Volume
5/24/2020	323.500000	324.239990	313.089996	318.250000	318.250000	93007000
5/17/2020	313.170013	320.890015	310.320007	318.890015	318.890015	133274700
5/10/2020	308.100006	319.690002	300.209991	307.709991	307.709991	208536900
5/3/2020	289.170013	310.350006	286.320007	310.130005	309.292755	168229000
4/26/2020	281.799988	299.000000	278.200012	289.070007	288.289612	197205100

Рисунок 3.1 – перші 5 елементів таблиці зі спостереженнями



Рисунок 3.2 – графік, що відображає динаміку ціни акцій компанії Apple

На графіку видно декілька значних «стрибків» росту акцій. Якщо провести невелике дослідження в Google, то можна помітити дисперсію між випуском або анонсом компанією нових продуктів, та ціною акцій, наприклад ріст ціни в кінці 2007 – початку 2008 можна зв’язати з випуском першого iPhone, ріст в кінці 2012 – випуск п’ятого iPhone, в травні 2015 року компанія презентувала оновлену лінійку MacBook. А в вересні 2018 компанія представила iPhone XS, після чого було досить помітне падіння ціни акцій майже на 75\$. В загалом просліджується

деяка річна та піврічна сезонність, росту чи падіння, що пов'язано з презентаціями нових продуктів.

3.4 Порівняльний аналіз моделей прогнозування

Спершу приведемо данні компанії Apple до потрібного вигляду та визначимо початкові параметри моделей. По перше розділимо ряд на дві частини: для тренування моделі та для тестування моделі (рис. 3.3). Я розділив їх у відношенні 90% і 10%.

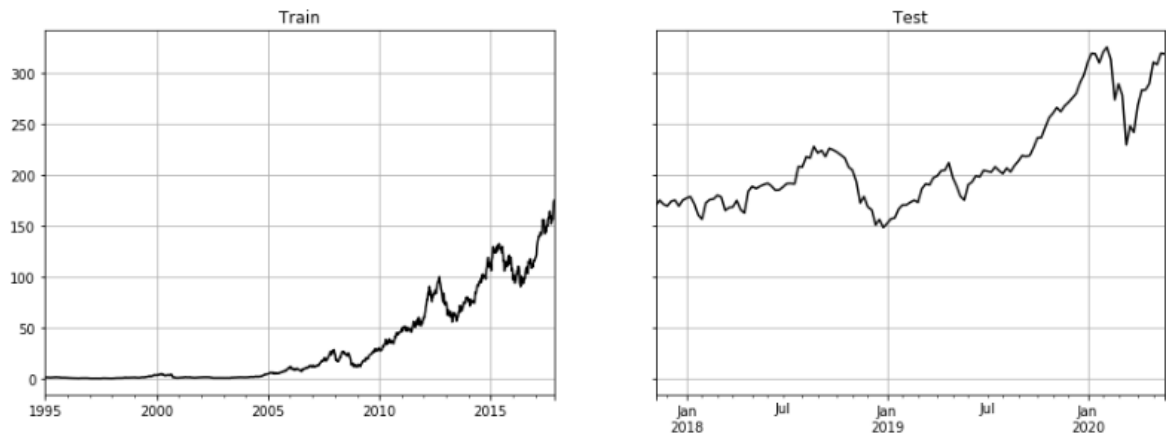


Рисунок 3.3 – дані, розбиті на train та test частини

Варто зазначити, що для моделювання ряду, на практиці використовують дещо модернізовану модель ARIMA, а саме SARIMAX, що має вигляд: $SARIMAX(p, d, q) \times (P, D, Q, s)$. Тут параметри P, D, Q є параметрами авторегресії, інтегрованості та ковзного середнього що відносяться до сезонної складової ряду. Параметр s вказує з якою частотою спостережень повторюється сезонний ефект. Декомпонуємо часовий ряд на трендову та сезонну частини (рис. 3.4). Провівши декілька досліджень, можна сказати що найменші залишки повертає сезонність частотою в 26 спостережень, тобто кожні пів року, тому використаємо цей параметр в побудові моделі.

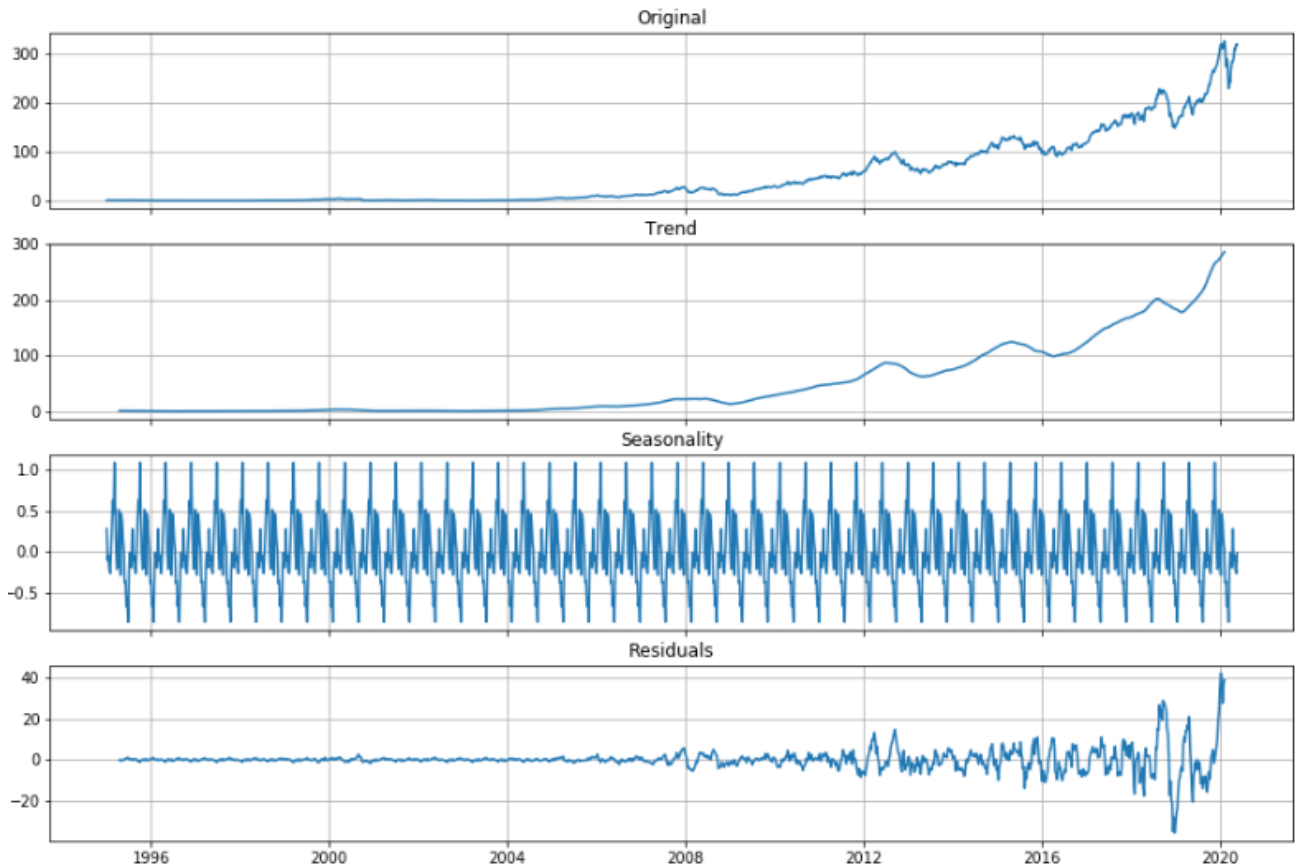


Рисунок 3.4 – дані, розбиті на train та test частини

1) Побудуємо модель $SARIMAX(p, d, q) \times (P, D, Q, 26)$. Для визначення параметрів p, d, q, P, D, Q використовуємо функцію *auto_arima* для автоматичного підбору. Результати роботи функції наведено на рис. 3.5.

SARIMAX Results						
Dep. Variable:	y	No. Observations:	1326			
Model:	SARIMAX(3, 2, 1)x(1, 0, 1, 26)		Log Likelihood	-3523.038		
Date:	Sat, 30 May 2020		AIC	7062.075		
Time:	01:52:19		BIC	7103.583		
Sample:	0		HQIC	7077.635		
- 1326						
Covariance Type:	opg					
	coef	std err	z	P> z	[0.025	0.975]
intercept	0.0011	0.001	1.475	0.140	-0.000	0.003
ar.L1	-0.0408	0.011	-3.849	0.000	-0.062	-0.020
ar.L2	0.0696	0.018	3.955	0.000	0.035	0.104
ar.L3	0.1286	0.008	15.379	0.000	0.112	0.145
ma.L1	-0.9960	0.005	-213.567	0.000	-1.005	-0.987
ar.S.L26	-0.5947	0.133	-4.467	0.000	-0.856	-0.334
ma.S.L26	0.6902	0.127	5.429	0.000	0.441	0.939
sigma2	11.9572	0.131	91.550	0.000	11.701	12.213
Ljung-Box (Q):	92.79	Jarque-Bera (JB):	137673.00			
Prob(Q):	0.00	Prob(JB):	0.00			
Heteroskedasticity (H):	1073.27	Skew:	-2.38			
Prob(H) (two-sided):	0.00	Kurtosis:	52.73			

Рисунок 3.5 – вибір найкращої SARIMAX моделі

Отже використаємо нашу модель SARIMAX (3,2,1) × (1,0,1,26) для тренування вхідних даних, та зрівняємо прогноз моделі з тестовими даними. Результат виведено на рис. 3.6. В таблиці 3.1 наведені метрики оцінювання моделі. Видно, що значення оцінки відносно хороші.

Таблиця 3.1. Метрики оцінювання моделі SARIMAX

	Error MEAN	Error STD	MAE	MAPE	MSE
Прогноз	-5	29	24	12%	885

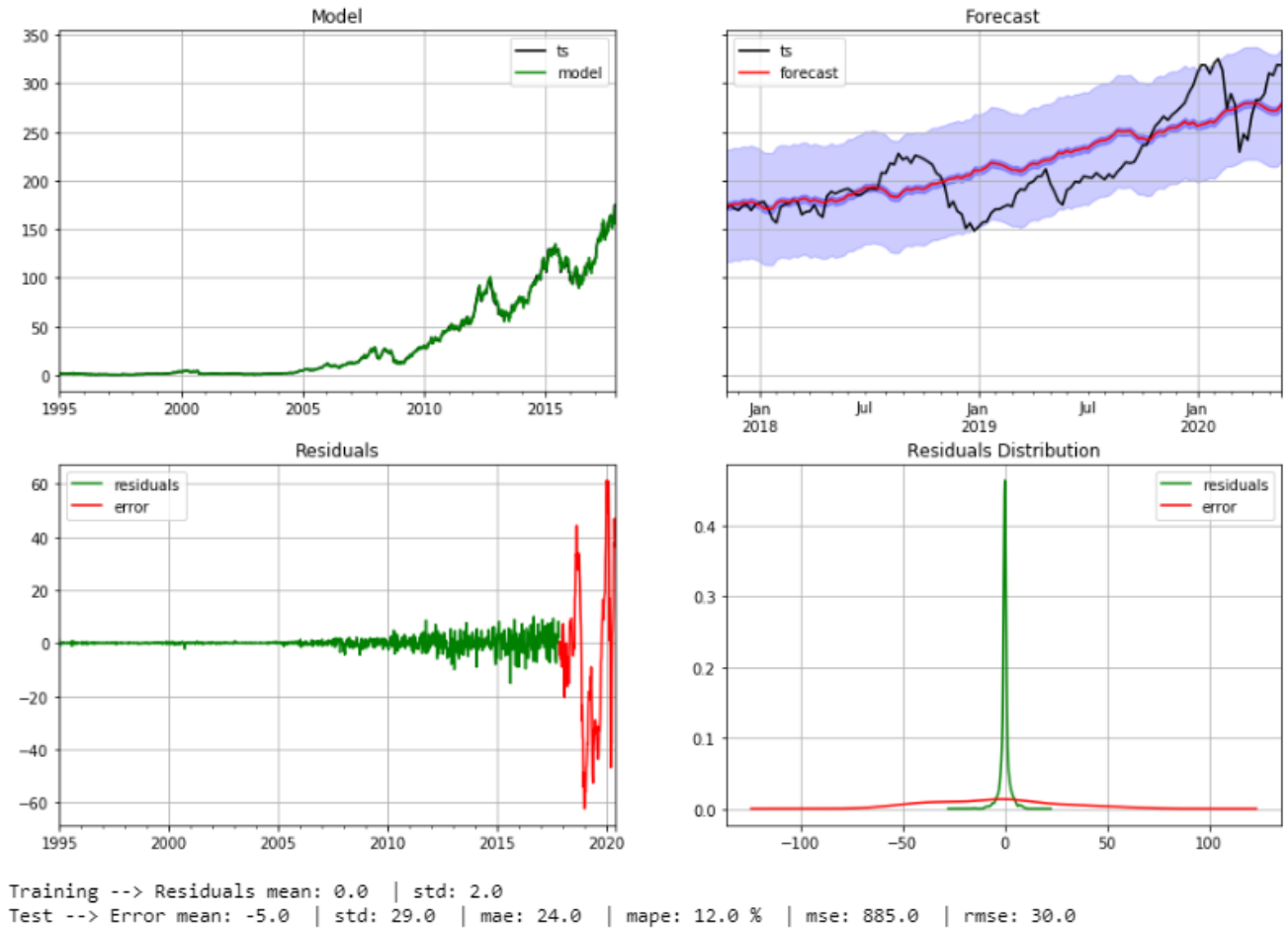


Рисунок 3.6 – результат роботи моделі SARIMAX

Видно що дана модель досить непогано апроксимує тренувальні данні, та будує гарне передбачення. Побудуємо передбачення на «невідомий» період, до кінця 2020 року (рис. 3.7). Отже, видно що ще до кінця року, згідно прогнозу, ціна акції перетне позначку в 360\$.

```
-- generating index date --> start: 2020-05-31 00:00:00 | end: 2020-12-27 00:00:00 | len: 31 ---
```

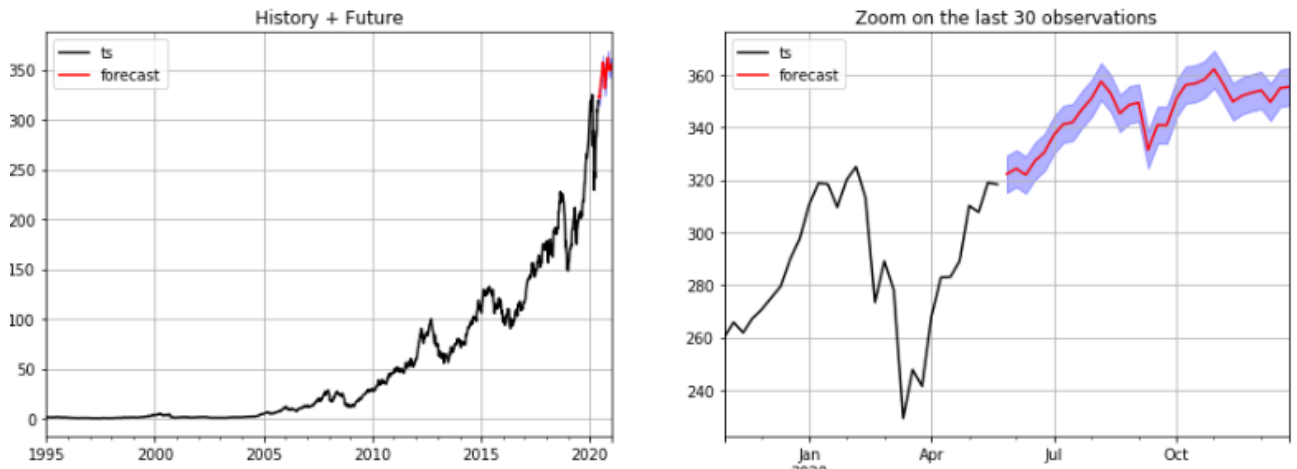


Рисунок 3.7 – результат прогнозування за допомогою моделі SARIMA

2) Модель, побудована за допомогою нейромережі LSTM приймає вхідний параметр, що відповідає розміру комірки довгострокової пам'яті. Найкращий прогноз дала модель з розміром пам'яті в 850 спостережень (рис. 3.8). Також метрики, наведені в таблиці 3.2, дуже схожі з метриками минулої моделі, і навіть трохи кращі. Тобто результат досить хороший.

Таблиця 3.2. Метрики оцінювання моделі LSTM

	Error MEAN	Error STD	MAE	MAPE	MSE
Прогноз	1	28	23	11%	773

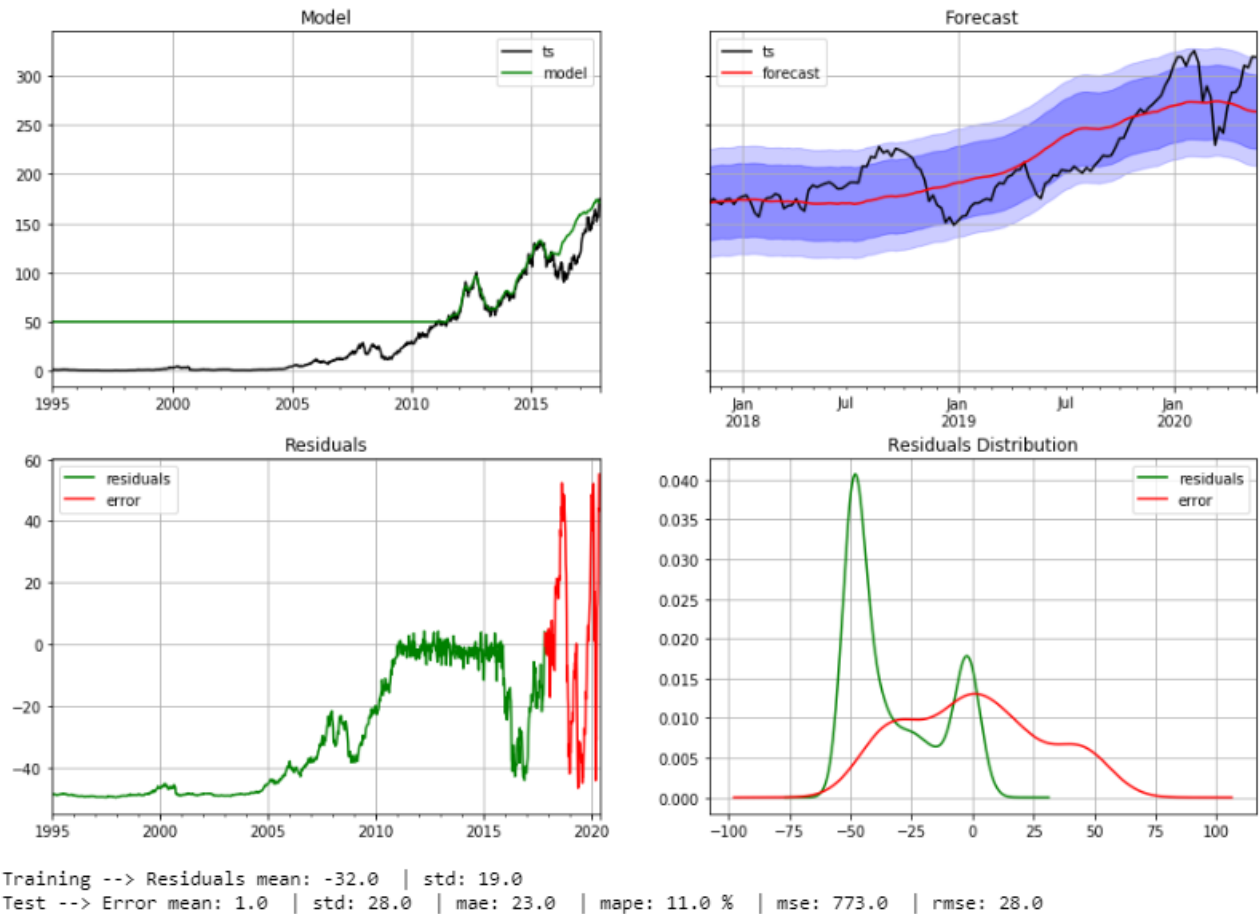


Рисунок 3.8 – результат роботи моделі LSTM

Побудуємо прогноз до кінця 2020 року за допомогою даної моделі (рис. 3.9). Видно що модель продовжила тенденцію до коливання моделі, і передбачує, що вартість акцій до кінця 2020 сягне відмітки в 500\$.

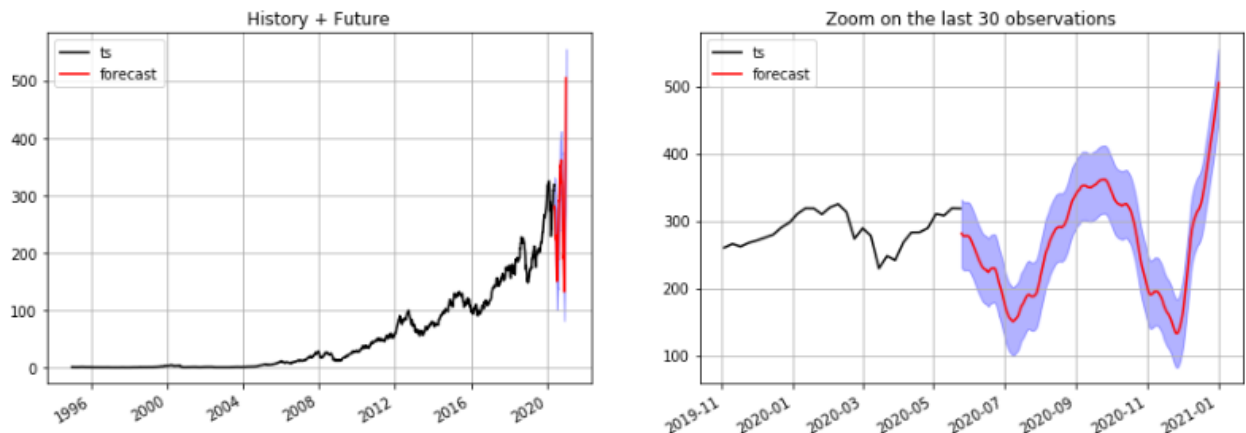


Рисунок 3.9 – результат прогнозування за допомогою моделі SARIMA

3) Prophet може автоматично будувати модель, що не завжди дає задовільні результати, тому додамо до моделі фактори щорічної та піврічної сезонності, а також список щорічних свят та подій, таких як новорічні свята, чорна п'ятниця, пасха, а також дні щорічних презентацій нових телефонів, що відбуваються приблизно на початку вересня. Модель наведено на рис. 3.10. Виходячи з графіків і значень оцінок (Таблиця 3.3), можна зрозуміти, що дана модель не дуже добре показує себе в прогнозуванні. Але видно, що модель гарно визначила тренд та сезонність, деякі скачки на графіку ігноруються як білий шум.

Таблиця 3.3. Метрики оцінювання моделі PROPHET

	Error MEAN	Error STD	MAE	MAPE	MSE
Прогноз	43	38	44	18%	3287

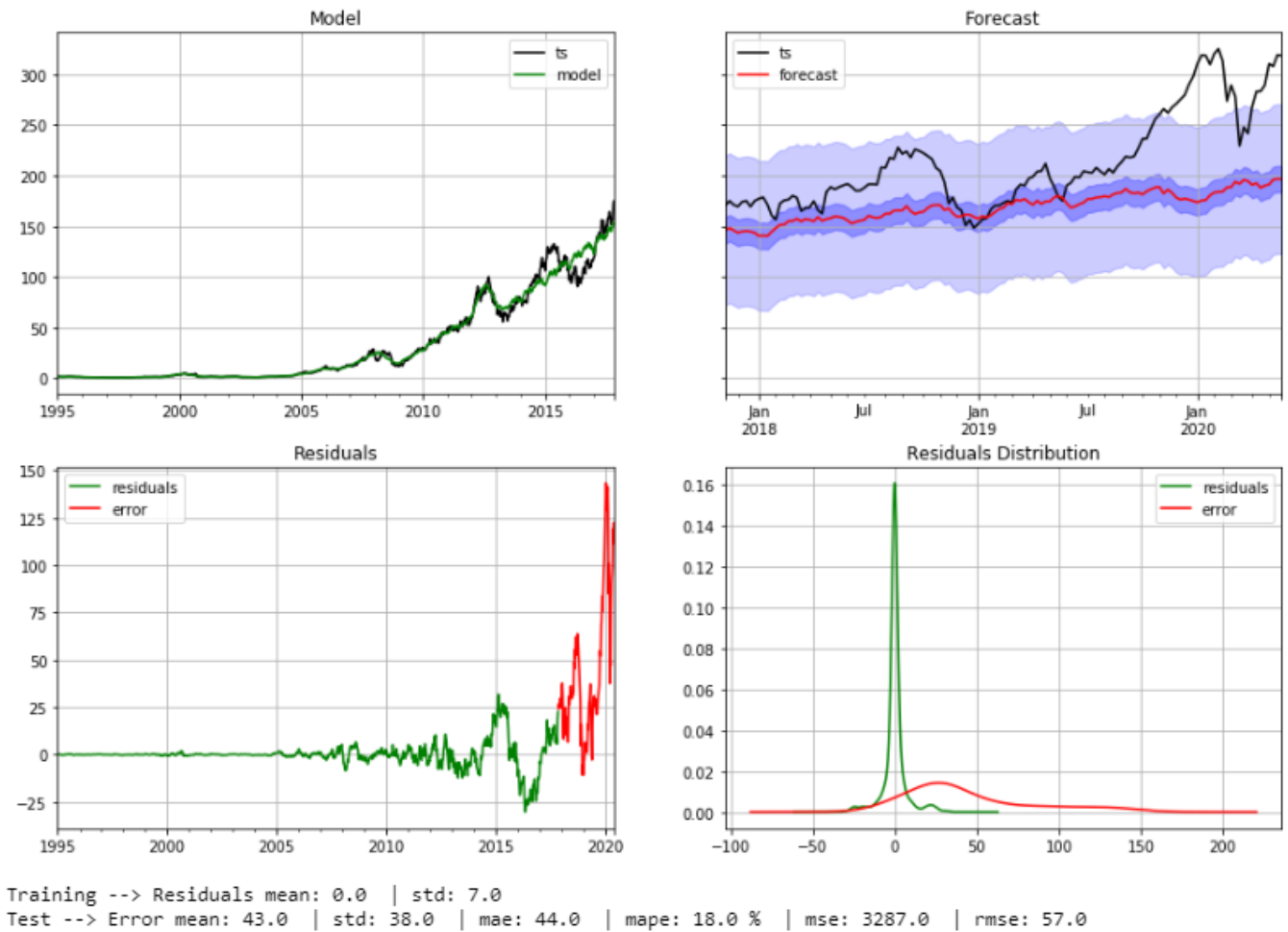


Рисунок 3.10 – результат роботи моделі Prophet

Також побудуємо прогноз за допомогою даної моделі (рис. 3.11). Модель фактично проігнорувала великі «стрибки» кінця 2019 року, та продовжує тенденцію, що утворилась на початку 2019 року. Модель прогнозує коливання цін на рівні 240-270 доларів до кінця року.

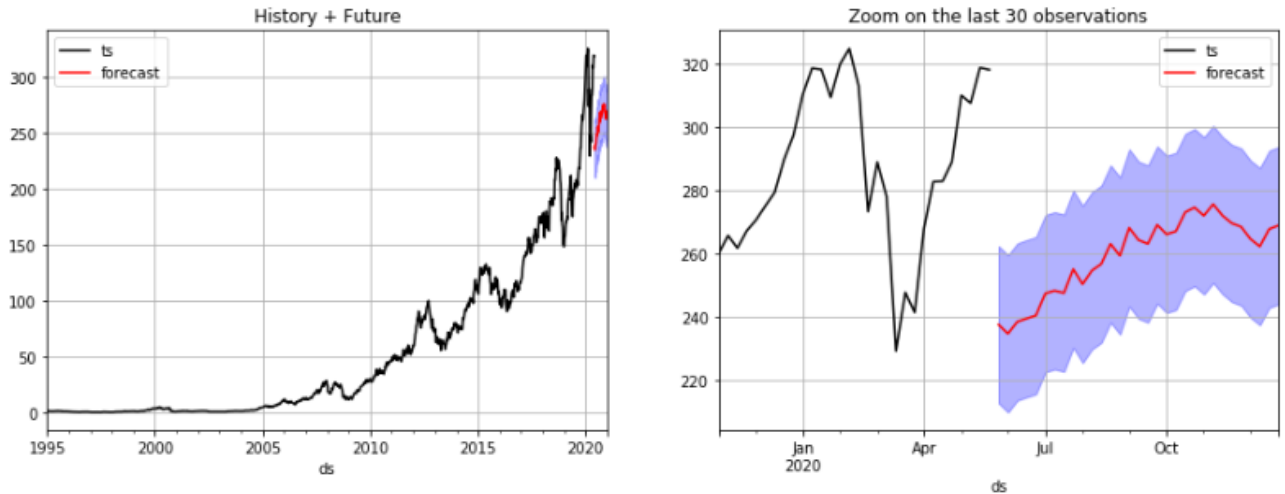


Рисунок 3.11 – результат роботи моделі Prophet

3.5 Висновки до розділу

Отже, оцінюючи графіки та метрики можна сказати що найкраще показали себе моделі, побудовані за допомогою LSTM та SARIMAX (таблиця 3.4). Що стосується графіків прогнозування на майбутнє, на мій погляд найправдоподібніший прогноз дає метод Prophet, адже після великих стрибків, як видно на графіку в період 2019 року, зазвичай бувають великі падіння, а тренд залишається сталим.

Таблиця 3.4. Метрики оцінювання трьох моделей

	Error MEAN	Error STD	MAE	MAPE	MSE
SARIMAX	-5	29	24	12%	885
LSTM	1	28	23	11%	773
PROPHET	43	38	44	18%	3287

РОЗДІЛ 4 ФУНКЦІОНАЛЬНО-ВАРТІСНИЙ АНАЛІЗ

4.1 Постановка завдання проектування

Проводиться оцінка основних характеристик програмного продукту, призначеного для прогнозування динаміки фондового ринку. Інтерфейс користувача був розроблений за допомогою мови програмування Python у середовищі розробки Jupiter Notebook. Програмний продукт призначено для використання на персональних комп'ютерах під управлінням операційних систем MacOS, Linux, Windows.

Нижче наведено аналіз різних варіантів реалізації модулю з метою вибору оптимальної, з огляду при цьому як на економічні фактори, так і на характеристики продукту, що впливають на продуктивність роботи і на його сумісність з апаратним забезпеченням. Для цього було використано апарат функціонально-вартісного аналізу.

4.2 Обґрунтування функцій та параметрів програмного продукту

Цільова функція F_0 - розробка програмного продукту, що реалізує прогнозування вартості цінних паперів за допомогою трьох методів моделювання. Вхідні данні це часовий ряд спостережень за ціною з конкретною частотою і в за деякий проміжок часу. Виходячи з конкретної задачі, можна виокремити наступні функції програмного продукту:

F_1 – Програмна реалізація:

а) мова програмування Python;

- б) мова програмування R;
- б) мова програмування Java;
- в) мова програмування MatLab.

F2 – Тип вхідних даних:

- а) часовий ряд вартості конкретної компанії;
- б) часовий ряд вартості фондового індексу.

F3 – реалізація графічного розв’язку задачі:

- а) використання вбудованих графічних методів;
- Б) власна реалізація графічного представлення.

Варіанти реалізації основних функцій наведено у морфологічній карті (Рис. 4.1)

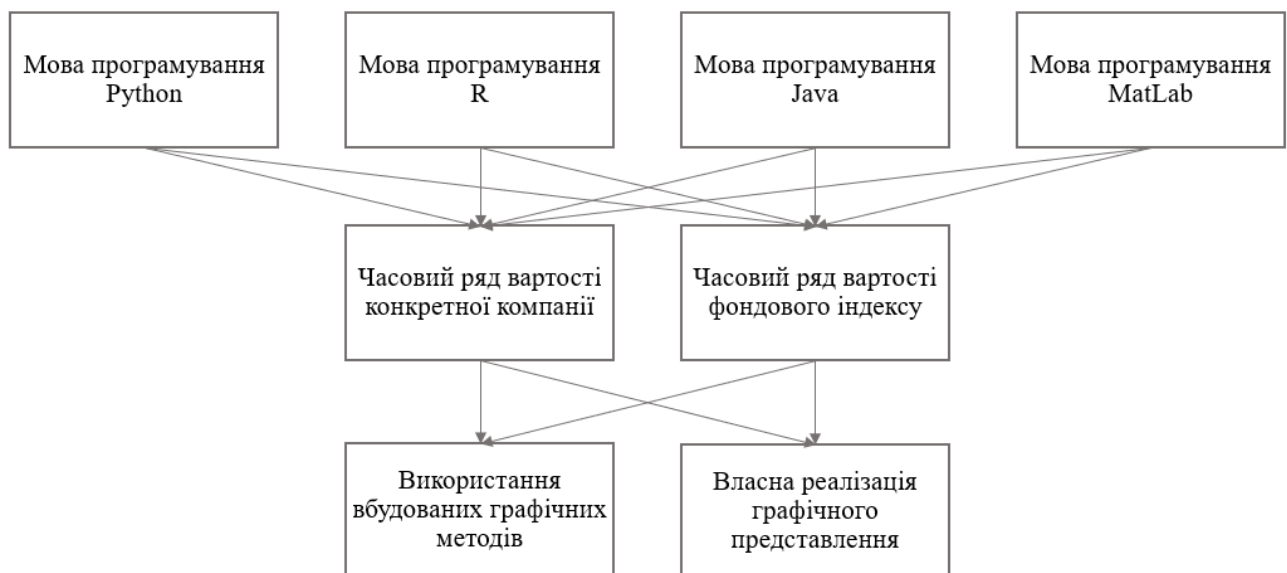


Рис. 4.1. Морфологічна карта.

Морфологічна карта показує всі можливі комбінації варіантів роботи функцій (їх реалізації). Дані функції складають у собі повну множину варіантів програмного продукту.

Використовуючи дану карту, було отримано матрицю варіантів основних функцій(Таблиця 4.1).

Таблиця 4.1. Позитивно-негативна карта.

Основні функції	Варіанти реалізації	Переваги	Недоліки
F1	а	Велика кількість готових методів, та методичних вказівок	Швидкість виконання методів, навчання нейронних мереж
	б	Більше аналітично направлених функцій	Важкий для вивчення, мало методичних вказівок
	в	Універсальна мова для більшості можливих задач	Не надто спеціалізований для побудови графіків та таблиць
	г	Зручний користувацький інтерфейс, гнучкий функціонал	Потребує дорогої ліцензії
F2	а	Практично націлений приклад з конкретними результатами	Велика частка білого шуму
	б	Індекс з широкоохоплюючим значенням, менша частка білого шуму	Не зрозуміле практичне використання
F3	а	Краща точність та швидкість побудови	Не надто гнучкі можливості

	б	Гнучкі можливості для створення дизайну, тощо	Необхідні сторонні бібліотеки та функції
--	---	---	--

На основі аналізу приведеної вище матриці, робимо висновок що деякі варіанти роботи були не відповідними до поставленої задачі та мають свої мінуси, а отже відповідні реалізації функцій необхідно відкинути, щоб зосередитися на правильних варіантах. Таким чином будемо використовувати наступні варіанти реалізації програмного продукту:

1. F1a – F2a – F3a;
2. F1a – F2б – F3a;
3. F1a – F2a – F3б;
4. F1a – F2б – F3б.

Для оцінювання якості розглянутих функцій обрана система параметрів, описана нижче.

Будемо розглядати три типи варіантів значення параметрів. Гірші, середні і кращі значення параметрів вибираються на основі вимог замовника й умов, що характеризують експлуатацію ПП як показано у таблиці 4.2.

Таблиця 4.2. Основні параметри програмного продукту.

Назва параметра	Умовні позначення	Одиниці виміру	Значення параметра		
			Гірші	Середні	Кращі

Час реалізації методу	X1	Год.	80	40	24
Час моделювання	X2	Хв	30	10	2
Значення середньої абсолютної відсоткової похибки (MAPE)	X3	%	40	20	5
Потенційний об'єм програмного коду	X4	кількість рядків коду	3000	2000	1200

За даними таблиці 4.2 будуються графічні характеристики параметрів – рис. 4.2 – рис. 4.5.

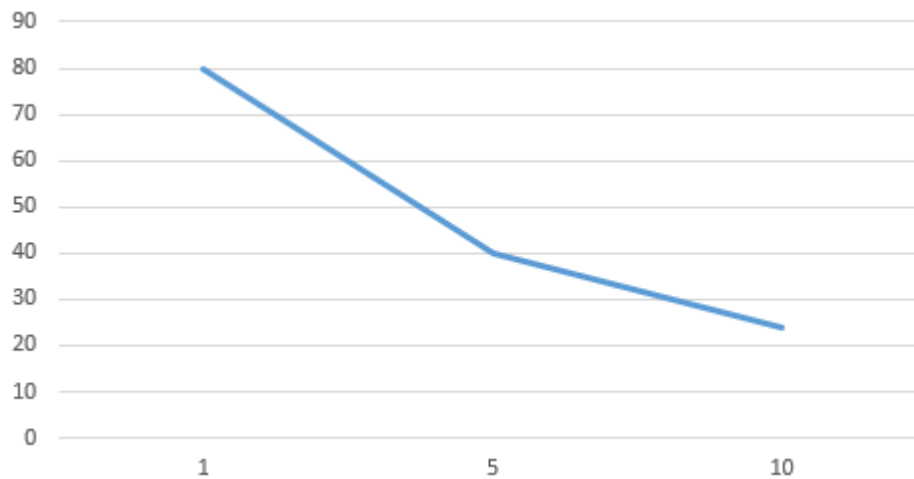


Рис. 4.2. X1, Час реалізації методу.

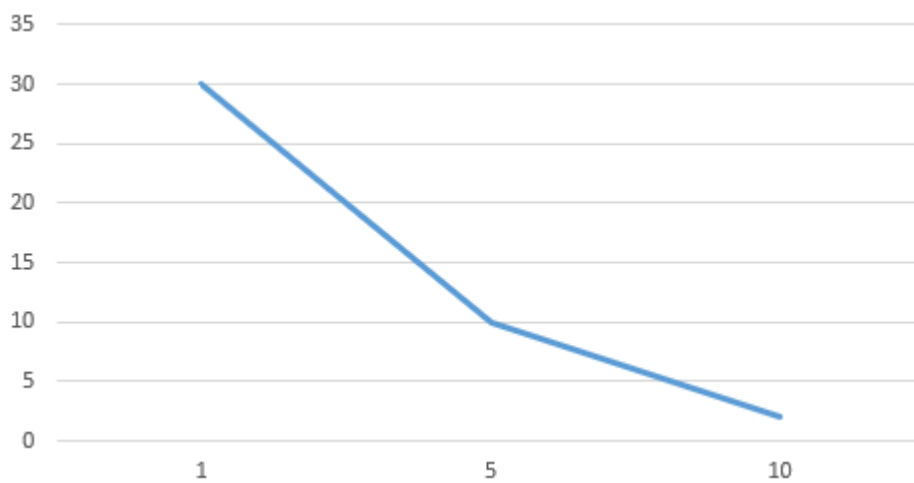


Рис. 4.3. X2, Час моделювання.

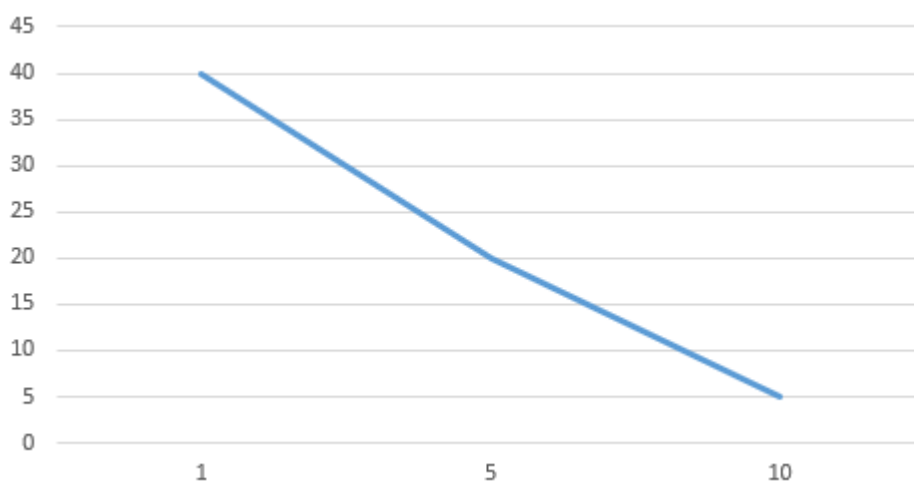


Рис. 4.4. X3, Значення середньої абсолютної відсоткової похибки (MAPE).

Порахуємо коефіцієнт узгодженості:

$$W = \frac{12S}{N^2(n^3-n)}, W = \frac{12*205}{(7^2*(4^3-4))} = \mathbf{0,84} > W_k = 0,67. \quad (4.5)$$

Ранжування можна вважати достовірним, тому що знайдений коефіцієнт узгодженості перевищує нормативний, котрий дорівнює 0,67.

Скориставшись результатами ранжування, проведемо попарне порівняння всіх параметрів і результати занесемо у таблицю 4.4, де найбільший ранг це 4, а найменший, відповідно, 1.

Таблиця 4.4. Попарне порівняння показників.

Параметри	Експерти							Кінцева оцінка	Числове значення
	1	2	3	4	5	6	7		
X1 та X2	>	>	>	>	>	<	>	>	1.5
X1 та X3	<	>	<	<	<	<	<	<	0.5
X1 та X4	>	>	>	>	>	>	>	>	1.5
X2 та X3	<	<	<	<	<	<	<	<	0.5
X2 та X4	>	>	>	>	<	>	>	>	1.5
X3 та X4	>	>	>	>	>	>	>	>	1.5

Таблиця 4.5. Розрахунок вагомості параметрів.

Параметри	Параметри				Перша ітер.		Друга ітер.		Третя ітер	
	X1	X2	X3	X4	b_i	K_{Bi}	b_i^1	K_{Bi}^1	b_i^2	K_{Bi}^2
X1	1,0	1,5	0,5	1,5	4,5	0.28125	16.25	0.275424	59.125	0.273727
X2	0,5	1,0	0,5	1,5	3,5	0.21875	12.25	0.207627	44.875	0.207755
X3	1,5	1,5	1,0	1,5	5,5	0.34375	21.25	0.360169	77.875	0.360532
X4	0,5	0,5	0,5	1,0	2,5	0.15625	9.25	0.15678	34.125	0.157986
Всього:					16	1.00	59.00	1.00	216.00	1.00

Як видно з таблиці, різниця значень коефіцієнтів вагомості не перевищує 2%, тому більшої кількості ітерацій не потрібно.

В даному ПП функція F1 описується критеріями X1 та X2, F2 - X3, F3 - X4.

Таблиця 4.6. Розрахунок показників рівня якості варіантів реалізації.

Основні функції	Варіант реалізації функції	Параметри	Абсолютне значення параметра	Бальна оцінка параметра	Коефіцієнт вагомості параметра	Коефіцієнт рівня якості
F1	А	X1	32	7.5	0.273	2,.48
		X2	8	6.25	0.207	1.294
F2	А	X3	12	7.7	0.361	2.778
	Б	X3	20	5	0.361	1.805
F3	А	X4	1200	10	0.157	1.57

	Б	X4	2200	3.75	0.157	0.589
--	---	----	------	------	-------	-------

Маємо наступні варіанти:

1. $F1(a) \Rightarrow F2(a) \Rightarrow F3(a)$
2. $F1(a) \Rightarrow F2(\bar{b}) \Rightarrow F3(a)$
3. $F1(a) \Rightarrow F2(a) \Rightarrow F3(\bar{b})$
4. $F1(a) \Rightarrow F2(\bar{b}) \Rightarrow F3(\bar{b})$

За даними з таблиці 4.6 визначаємо рівень якості кожного з варіантів:

$$K_{я1} = 2.048 + 1.294 + 2.778 + 1.57 = \mathbf{7.69}$$

$$K_{я2} = 2.048 + 1.294 + 1.805 + 1.57 = \mathbf{6.717}$$

$$K_{я3} = 2.048 + 1.294 + 2.778 + 0.589 = \mathbf{6.709}$$

$$K_{я4} = 2.048 + 1.294 + 1.805 + 0.589 = \mathbf{5.736}$$

Отже, найкращим є перший варіант, для якого коефіцієнт технічного рівня має найбільше значення.

4.3 Економічний аналіз варіантів розробки ПП

Для оцінки трудомісткості розробки спочатку проведемо розрахунок трудомісткості. Усі варіанти мають наступні основні завдання:

- 1) програмна реалізація методів;
- 2) розробка оболонки програмного продукту;

Також кожний з варіантів має два додаткових завдання, які є реалізаціями розгалужених варіантів розробки незалежного модуля. Далі наведено варіанти додаткових завдань (два завдання, які мають номери 3 в реалізаціях та два завдання, які мають номери 4 в реалізаціях)

3.1) вхідні дані в вигляді часового ряду вартості конкретної компанії;

3.2) вхідні дані в вигляді часового ряду вартості фондового індексу;

4.1) використання вбудованих графічних методів;

4.2) власна реалізація графічного представлення.

В варіанті 1 присутні наступні додаткові завдання під номерами 3.1 та 4.1

В варіанті 2 присутні наступні додаткові завдання під номерами 3.2 та 4.1

В варіанті 3 присутні наступні додаткові завдання під номерами 3.1 та 4.2

В варіанті 4 присутні наступні додаткові завдання під номерами 3.2 та 4.2

За ступенем новизни до групи А відносяться завдання 1 та 4.2;

До групи Б відносяться завдання 4.1 та 3.1;

До групи В відноситься завдання 3.2;

До групи Г відноситься завдання 2;

За складністю алгоритмів до групи 1 відносяться завдання 1 та 4.2;

До групи 2 відноситься завдання 4.1;

До групи 3 відносяться завдання 3.2, 3.1 та 2;

Якщо при розробці ПП використовуються стандартні бібліотеки та пакети прикладних програм, норми часу коректуються з допомогою коефіцієнта $K_{ст}$, значення якого лежать в діапазоні 0,6-0,8. Для даної задачі візьмемо $K_{ст} = 0,7$.

Для першого завдання (ступінь новизни А, група складності – 1), тобто $T_p = 90$ людино-днів, $K_{п} = 1,7$, $K_{СК} = 1$; $K_{СТ} = 0.7$. Тоді, загальна трудомісткість програмування першого завдання дорівнює:

$$T_1 = 90 \cdot 1,7 \cdot 1 \cdot 0,7 = 107,1 \text{ людино-днів}$$

Для другого завдання (ступінь новизни Г, група складності – 2), тобто $T_p = 12$ людино-днів, $K_{п} = 0,43$; $K_{СК} = 1$; $K_{СТ} = 0.7$:

$$T_2 = 12 \cdot 0,43 \cdot 1 \cdot 0,7 = 3,62 \text{ людино-днів.}$$

Для завдання 3.1 (ступінь новизни Б, група складності – 3), тобто $T_p = 19$ людино-днів, $K_{п} = 0,9$; $K_{СК} = 1$; $K_{СТ} = 0.7$:

$$T_{3.1} = 12 \cdot 0,6 \cdot 1 \cdot 0,7 = 11,97 \text{ людино-днів.}$$

Для завдання 3.2 (ступінь новизни В, складність алгоритмів – 3), тобто $T_p = 12$ людино-днів, $K_{п} = 0,6$; $K_{СК} = 1$; $K_{СТ} = 0.7$

$$T_{3.2} = 12 \cdot 0,6 \cdot 1 \cdot 0,7 = 5,04 \text{ людино-днів.}$$

Для завдання 4.1 (ступінь новизни Б, група складності – 2), тобто $T_p = 27$ людино-днів, $K_{п} = 1,08$; $K_{СК} = 1$; $K_{СТ} = 0.7$:

$$T_{4.1} = 27 \cdot 1,08 \cdot 1 \cdot 0,7 = 20,41 \text{ людино-днів.}$$

Для завдання 4.2 (ступінь новизни А, група складності – 1), тобто $T_p = 90$ людино-днів, $K_{п} = 1,7$, $K_{СК} = 1$; $K_{СТ} = 0.7$.

$$T_{4.2} = 90 \cdot 1,7 \cdot 1 \cdot 0,7 = 107,1 \text{ людино-днів}$$

Складаємо трудомісткість відповідних завдань для кожного з обраних варіантів реалізації програми, щоб отримати їх трудомісткість:

$$T_I = (107.1 + 3.62 + 5.04 + 20.41) \cdot 8 = 1,089.36 \text{ людино-годин};$$

$$T_{II} = (107.1 + 3.62 + 11.97 + 20.41) \cdot 8 = 1,144.8 \text{ людино-годин.}$$

$$T_{III} = (107.1 + 3.62 + 5.04 + 107.1) \cdot 8 = 1,782.88 \text{ людино-годин.}$$

$$T_I = (107.1 + 3.62 + 11.97 + 107.1) \cdot 8 = 1,838.32 \text{ людино-годин.}$$

Найбільш високу трудомісткість має варіант IV.

В розробці беруть участь два програмісти з окладом **13300** грн., один фінансовий аналітик з окладом **15960** грн. Визначимо зарплату за годину:

$$CЧ = \frac{13300 + 13300 + 15960}{3 \cdot 21 \cdot 8} = \mathbf{84.45 \text{ грн.}}$$

Тоді, заробітну плату розробників за варіантами становить:

$$I. \quad CЗП = 84.45 \cdot 1.089.36 \cdot 1.2 = 110,395.74 \text{ грн.};$$

$$II. \quad CЗП = 84.45 \cdot 1.144.8 \cdot 1.2 = 116,014.03 \text{ грн.}$$

$$III. \quad CЗП = 84.45 \cdot 1.782.88 \cdot 1.2 = 180,677.06 \text{ грн.}$$

$$IV. \quad CЗП = 84.45 \cdot 1.838.32 \cdot 1.2 = 186,295.35 \text{ грн.}$$

Відрахування на єдиний соціальний внесок становить 22%:

$$I. \quad СВІД = CЗП \cdot 0.22 = 77985.63 \cdot 0.22 = 24,287.06 \text{ грн.}$$

$$II. \quad СВІД = CЗП \cdot 0.22 = 80537.81 \cdot 0.22 = 25,523.09 \text{ грн.}$$

$$III. \quad СВІД = CЗП \cdot 0.22 = 167141.8 \cdot 0.22 = 39,748.95 \text{ грн.}$$

$$IV. \quad СВІД = CЗП \cdot 0.22 = 169695.39 \cdot 0.22 = 40,984.98 \text{ грн.}$$

Тепер, визначимо витрати на оплату однієї машино-години. (C_M)

Так як одна ЕОМ обслуговує одного програміста з окладом 13300 грн., з коефіцієнтом зайнятості 0,2 то для однієї машини отримаємо:

$$C_M = 12 \cdot M \cdot K_3 = 12 \cdot 13300 \cdot 0,2 = 31,920 \text{ грн.}$$

З урахуванням додаткової заробітної плати:

$$C_{3П} = C_M \cdot (1 + K_3) = 31,920 \cdot (1 + 0,2) = 38,304 \text{ грн.}$$

Відрахування на єдиний соціальний внесок:

$$C_{ВІД} = C_{3П} \cdot 0,22 = 38,304 \cdot 0,22 = 8,426.88 \text{ грн.}$$

Амортизаційні відрахування розраховуємо при амортизації 25% та вартості ЕОМ – **14999** грн.

$$C_A = K_{TM} \cdot K_A \cdot Ц_{ПР} = 1.15 \cdot 0.25 \cdot 14999 = 4,312.21 \text{ грн.,}$$

Витрати на ремонт та профілактику розраховуємо як:

$$C_P = K_{TM} \cdot Ц_{ПР} \cdot K_P = 1.15 \cdot 14999 \cdot 0.05 = 862.44 \text{ грн.,}$$

Ефективний годинний фонд часу ПК за рік розраховуємо за формулою:

$$T_{ЕФ} = (D_K - D_B - D_C - D_P) \cdot t_3 \cdot K_B = (365 - 104 - 8 - 16) \cdot 8 \cdot 0.9 = 1706.4 \text{ год.,}$$

Витрати на оплату електроенергії розраховуємо за формулою:

$$C_{ЕЛ} = T_{ЕФ} \cdot N_C \cdot K_3 \cdot Ц_{ЕН} = 1706.4 \cdot 0,84 \cdot 0,81 \cdot 1,75 = 2,031.81 \text{ грн.,}$$

Накладні витрати розраховуємо за формулою:

$$C_H = Ц_{ПР} \cdot 0.67 = 14999 \cdot 0,67 = 10,049.33 \text{ грн.}$$

Тоді, річні експлуатаційні витрати будуть:

$$C_{ЕКС} = C_{3П} + C_{ВІД} + C_A + C_P + C_{ЕЛ} + C_H$$

$$C_{\text{ЕКС}} = 38,304 + 8,426.88 + 4,312.21 + 862.44 + 2,031.81 + 10,049.33 = 63,986.67 \text{ грн.}$$

Собівартість однієї машино-години ЕОМ дорівнюватиме:

$$C_{\text{М-Г}} = C_{\text{ЕКС}} / T_{\text{ЕФ}} = 63,986.67 / 1706.4 = 37,5 \text{ грн/час.}$$

Так як в обидвох варіантах всі роботи, які пов'язані з розробкою програмного продукту ведуться на ЕОМ, витрати на оплату машинного часу, в залежності від обраного варіанта реалізації, складає:

$$C_{\text{М}} = C_{\text{М-Г}} \cdot T$$

$$\text{I. } C_{\text{М}} = 84.45 \cdot 1,089.36 = 91,996.45 \text{ грн.}$$

$$\text{II. } C_{\text{М}} = 84.45 \cdot 1,144.8 = 96,678.36 \text{ грн.}$$

$$\text{III. } C_{\text{М}} = 84.45 \cdot 1,782.88 = 150,564.21 \text{ грн.}$$

$$\text{IV. } C_{\text{М}} = 84.45 \cdot 1,838.32 = 155,246.12 \text{ грн.}$$

Накладні витрати складають 67% від заробітної плати:

$$C_{\text{Н}} = C_{\text{ЗП}} \cdot 0,67$$

$$\text{I. } C_{\text{Н}} = 110,395.74 \cdot 0.67 = 73,965.14 \text{ грн.}$$

$$\text{II. } C_{\text{Н}} = 116,014.03 \cdot 0.67 = 77,729.4 \text{ грн.}$$

$$\text{III. } C_{\text{Н}} = 180,677.06 \cdot 0.67 = 121,053.63 \text{ грн.}$$

$$\text{IV. } C_{\text{Н}} = 186,295.35 \cdot 0.67 = 124,817.88 \text{ грн.}$$

Отже, вартість розробки ПП за варіантами становить:

$$C_{\text{ПП}} = C_{\text{ЗП}} + C_{\text{Від}} + C_{\text{М}} + C_{\text{Н}}$$

$$\text{I. } C_{\text{ПП}} = 110,395.74 + 24,287.06 + 91,996.45 + 73,965.14 = 300,644.39 \text{ грн.};$$

II. $C_{\text{ПП}} = 116,014.03 + 25,523.09 + 96,678.36 + 77,729.4 = 315,944.88$ грн.

III. $C_{\text{ПП}} = 180,677.06 + 39,748.95 + 150,564.21 + 121,053.63 = 492,043.85$ грн.

IV. $C_{\text{ПП}} = 186,295.35 + 40,984.98 + 155,246.12 + 124,817.88 = 507,344.33$ грн.

Вибір кращого варіанта ПП техніко-економічного рівня.

Розрахуємо коефіцієнт техніко-економічного рівня за формулою:

$$K_{\text{ТЕР}j} = K_{kj} / C_{\text{ПП}j},$$

$$K_{\text{ТЕР}1} = 7.69 / 300,644.39 = \mathbf{2.56 \cdot 10^{-5}}$$

$$K_{\text{ТЕР}2} = 6.717 / 315,944.88 = 2.126 \cdot 10^{-5}$$

$$K_{\text{ТЕР}3} = 6.709 / 492,043.85 = 1.36 \cdot 10^{-5}$$

$$K_{\text{ТЕР}4} = 5.736 / 507,344.33 = 1.13 \cdot 10^{-5}$$

Як бачимо, найбільш ефективним є перший варіант реалізації програми з коефіцієнтом техніко-економічного рівня $K_{\text{ТЕР}1} = \mathbf{2.56 \cdot 10^{-5}}$.

4.4 Висновки до розділу 4

Після виконання функціонально-вартісного аналізу програмного комплексу, що розроблюється, можна зробити висновок, що з альтернатив, що залишились після першого відбору двох варіантів виконання програмного комплексу оптимальним є перший варіант реалізації програмного продукту. У нього виявився найкращий показник техніко-економічного рівня якості:

$$K_{\text{ТЕР}1} = 7.69 / 300,644.39 = \mathbf{2.56 \cdot 10^{-5}}$$

До першого варіанту реалізації відносяться такі задачі:

- програмна реалізація за допомогою мови Python;
- часовий ряд вартості конкретної компанії;
- використання вбудованих графічних методів;

Як бачимо, даний варіант реалізації дозволяє швидко та просто реалізувати необхідні функції, а також задовольняє потреби у функціональності.

ВИСНОВКИ

Отже, в цій роботі були дослідженні основні напрямки та методи прогнозування, що використовуються на фондових ринках. Також було проаналізовано увесь фондовий ринок, як частину світової економіки. Проведені дослідження, що базуються на моделюванні часових рядів, дають можливість передбачити значення вартості на деякий майбутній крок часу з деякою точністю.

ПЕРЕЛІК ПОСИЛАНЬ:

1. ЗАКОН УКРАЇНИ Про цінні папери та фондовий ринок. URL: <https://zakon.rada.gov.ua/cgi-bin/laws/main.cgi?nreg=3480-15> (дата звернення 05.04.2020)
2. СЛОВНИК ФІНАНСОВО-ПРАВОВИХ ТЕРМІНІВ 2-ге видання, перероблене і доповнене, 2012. URL: <https://web.archive.org/web/20160509113740/http://ndifp.asta.edu.ua/files/doc/slovyk-finpravo.pdf> (дата звернення 05.04.2020)
3. Edwin LeFevre The Sun Dial Press, REMINISCENCES OF A STOCK OPERATOR, Garden City, New York Copyright 1923. 269p.
4. How the Stock Market Works. URL: <https://www.thebalance.com/how-does-the-stock-market-work-3306244> (дата звернення 22.04.2020)
5. Stock Trading Software. URL: <https://www.marketreview.com/stocks-trading-software/> (дата звернення 24.04.2020)
6. Как устроена фондовая биржа. URL: <https://fincult.info/article/kak-ustroena-fondovaya-birzha/> (дата звернення 26.04.2020)
7. Как происходят торги на торговой площадке биржи. URL: <https://berg.com.ua/basic/trading-floor/> (дата звернення 26.04.2020)
8. John J. Myrphy, Technical analysis of the financial markets, New York Institute of Finance 1985. 542p.
9. Biggest Stock Market Crashes Throughout History. URL: <https://www.thestreet.com/markets/history-of-stock-market-crashes-14702941> (дата звернення 27.04.2020)
10. Лукашин Ю. П. Адаптивні методи короткострокового прогнозування часових рядів: навч. посіб. \ Москва: Инфра-М, 2003. 416 с.
11. Box G., Jenkins G. Time Series Analysis: Forecasting and Control. / San

- Francisco: Holden-Day, 1976. 420 с.
12. Бідюк П.І., Аналіз часових рядів: навч. посіб. / Київ: НТУУ «КПІ», 2013. 600 с.
 13. Бідюк П. І., Часові ряди: моделювання і прогнозування: монографія / Київ: ЕКМО, 2003. 144 с.
 14. Sepp Hochreiter, LONG SHORT-TERM MEMORY Neural Computation 9(8), Technische Universität München, Germany 1997, 35с.
 15. Sean J. Taylor, Benjamin Letham, Forecasting at Scale 2017, 26p.
 16. Hyndman, R.J., & Athanasopoulos, G., Forecasting: principles and practice, 2nd edition, OTexts: Melbourne, Australia, 2018. 241p.
 17. Lufuno Ronald Marwala, Forecasting the Stock Market Index Using Artificial Intelligence Techniques, Johannesburg 2006. 146p.
 18. ЕГОРОВА Н. Е., БАХТИЗИН А. Р., ТОРЖЕВСКИЙ К. А., ПРОГНОЗИРОВАНИЕ ФОНДОВЫХ РЫНКОВ С ИСПОЛЬЗОВАНИЕМ ЭКОНОМИКО-МАТЕМАТИЧЕСКИХ МОДЕЛЕЙ, Москва: URSS 2013г. 216р.
 19. Афанасьев В. Н., Юзбашев М. М., Анализ временных рядов и прогнозирование: учебник. Москва, МГУ 2001. 228 с.
 20. Бердникова Т. Б., Рынок ценных бумаг и биржевое дело: учебное пособие. Москва: ИНФРА-М, 2002. 207 с.
 21. Первозванский А. А., Первозванская Т. Н., Финансовый рынок: расчет и риск. Москва: Инфра-М, 1994. 192 с.
 22. Жукова Е. Ф., Рынок ценных бумаг: учебник для студентов и вузов, обучающихся по экономическим специальностям 3-е изд., перераб.и доп. Москва: Инфра-М, 2009. 567 с.
 23. Галанова В.А., Басова А.И. Рынок ценных бумаг: -2-е изд., перераб. И доп., Москва: Инфра-М, 2002. 448 с.

ДОДАТОК А ІЛЮСТРАТИВНИЙ МАТЕРІАЛ

Порівняльний аналіз методів прогнозування фондового ринку

Виконав: Русин Данило Костянтинович
Науковий керівник: проф. Зайченко О. Ю.

Актуальність роботи

- Фінансовий ринок це найважливіша частина світової економіки
- Вміння аналізувати та прогнозувати динаміку вартості цінних паперів надає широкі можливості в брокерській сфері
- Методи, що використовуються в даній роботі- універсальні, та можуть бути застосовані для моделювання часових рядів та графіків в будь якій сфері

Мета, об'єкт та предмет роботи

- **Мета дослідження** - проаналізувати існуючі методи моделювання та прогнозування та реалізувати їх на вхідних даних
- **Предмет дослідження** - методи моделювання ARIMA, LSTM, PROPHET
- **Об'єкт дослідження** - набір спостережень за вартістю цінних паперів конкретної компанії в конкретний період часу

Постановка задачі

- Проаналізувати існуючі літературні джерела, що присвячені технічному та фундаментальному аналізу фондового ринку
- Дослідити математичні методи, що найкраще підходять для задач моделювання та прогнозування
- Побудувати програмну реалізацію методів прогнозування
- Провести експерименти на вхідних даних для вибору найефективнішого методу
- Порівняти отримані результати

Проблематика

- На вартість цінних паперів мають вплив соціально-економічні, політичні, соціальні та інформаційні чинники, що набагато ускладнює або ж навіть унеможлиблює прогнозування на довготривалий період

Моделі часових рядів

- Для прогнозування майбутніх значень часового ряду необхідно спершу ідентифікувати його компоненти.
- Складові: тренд, сезонність, білий шум.
- Часовий ряд можна представити як суму або добуток тренду, сезонної і випадкової компонент.

Процес авторегресії та ковзного середнього

- **AR(p)** $x_t = \xi + \phi_1(x_{t-1}) + \phi_2(x_{t-2}) + \phi_3(x_{t-3}) + \dots + \varepsilon$

Де, ξ - вільний член, а $\bar{\phi}_{1,n}$ - параметри авторегресії.

- **MA(q)** $x_t = \mu + \varepsilon_1 - \theta_1\varepsilon_{t-1} - \theta_2\varepsilon_{t-2} - \theta_3\varepsilon_{t-3} - \dots$

Де, μ це константа, а $\bar{\theta}_{1,n}$ - параметри ковзного середнього.

- **ARMA(p,q)** $x_t = c + \varepsilon_t + \sum_{i=1}^p \alpha_i X_{t-i} + \sum_{i=1}^q \beta_i \varepsilon_{t-i}$

Де, c - константа, ε_t - білий шум, а α_i та β_i це авторегресійні коефіцієнти та коефіцієнти ковзного середнього відповідно.

Модель ARIMA(p,d,q)

$$\Delta^d X_t = c + \varepsilon_t + \sum_{i=1}^p \alpha_i \Delta^d X_{t-i} + \sum_{j=1}^q \beta_j \varepsilon_{t-j}$$

Де ε_t це стаціонарний часовий ряд, c, α_i, β_j - параметри моделі, а Δ^d - оператор різниці часового ряду порядку d .

Етапи побудови моделі:

- Ідентифікація параметрів моделі
- Оцінка параметрів моделі
- Оцінка самої моделі

LSTM нейронні мережі

Основна ідея даного методу це виділення окремої комірки пам'яті, для зберігання інформації, отриманої в минулих моментах часу. Для контролю цієї комірки використовуються три «вентилі»:

- Вентиль пам'яті f_t
- Вхідний вентиль i_t
- Вихідний вентиль o_t

Алгоритм навчання LSTM

$$\begin{aligned} f_t &= \sigma_g(W_f x_t + U_f h_{t-1} + b_f) \\ i_t &= \sigma_g(W_i x_t + U_i h_{t-1} + b_i) \\ o_t &= \sigma_g(W_o x_t + U_o h_{t-1} + b_o) \\ h_t &= f_t \circ c_{t-1} + f_t \circ \sigma_c(W_c x_t + U_c h_{t-1} + b_c) \\ y_t &= o_t \circ \sigma_c(h_t) \end{aligned}$$

Де: x_t - вектор вхідних даних;

y_t - вектор вихідних даних;

h_t - вектор стану комірки пам'яті;

W, U та b - матриці та вектор параметрів;

f_t, i_t, o_t - вектори вентилів;

σ - це функція активації, σ_g - сигмоїдна функція, σ_c - функція гіперболічного тангенсу.

Prophet

В основі методології лежить процедура підгонки адитивних регресійних моделей наступного вигляду:

$$y(t) = g(t) + s(t) + h(t) + \epsilon_t$$

Де, $g(t)$ - тренд функція, $s(t)$ - сезонні коливання, $h(t)$ - функція, що відображає ефекти святкових днів та інших впливових подій, ϵ_t - нормально розподілені випадкові збурення.

- Тренд моделюється за допомогою кусково-лінійної регресії або логістичної кривої росту.
- Річна та тижнева сезонність моделюється за допомогою рядів Фур'є.
- «Свята» представлені у вигляді індикаторних змінних.

Оцінка моделей

Дано: $x_1, \dots, x_n \sim N(\mu, \sigma^2)$ з середнім значенням вибірки $\bar{x} = \frac{x_1 + \dots + x_n}{n}$ що є випадковою змінною, розподіленою таким чином, що $\bar{x} \sim N(\mu, \frac{\sigma^2}{n})$. Тоді статистичні похибки є $e_i = x_i - \mu$, а залишки це $r_i = x_i - \bar{x}$.

Тоді:

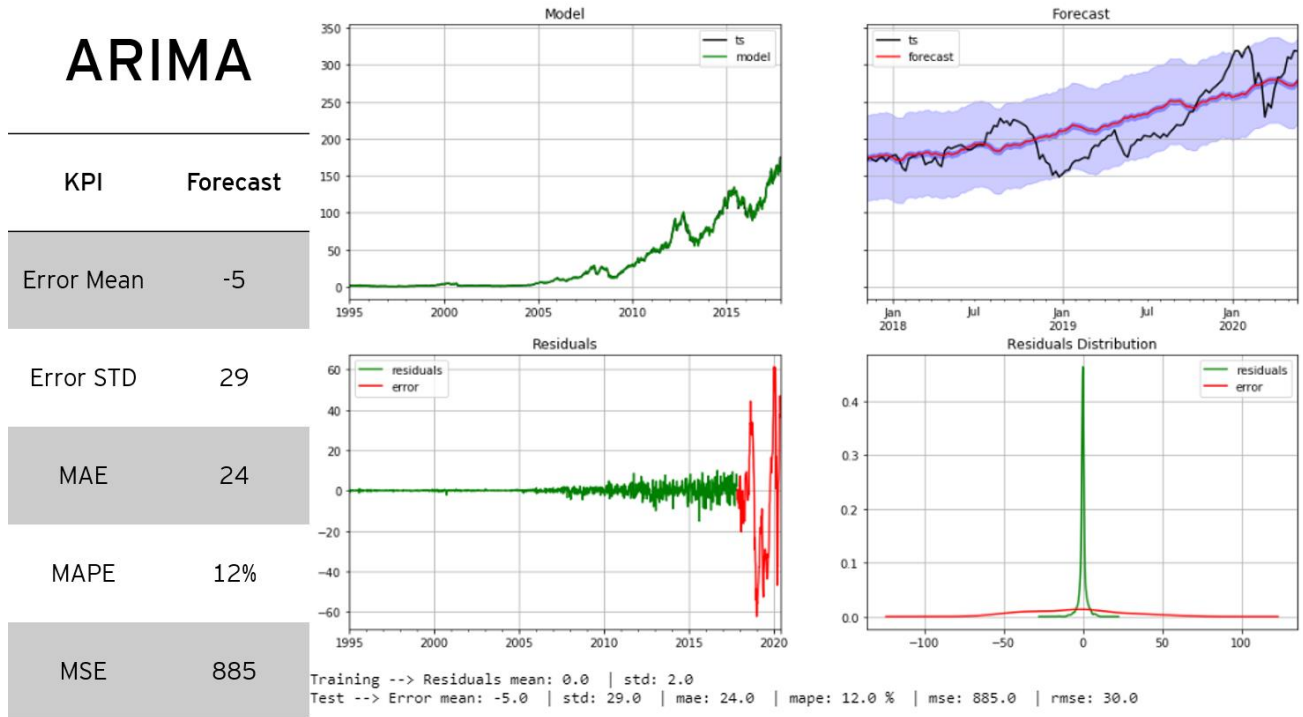
- $STD = \sigma = \sqrt{\sigma^2} = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - \bar{x})^2}$
- $MSE = \frac{1}{N} \sum_{i=1}^N (x_i - \bar{x})^2$
- $MAE = \frac{1}{N} \sum_{i=1}^N |x_i - \bar{x}|$
- $MAPE = \frac{1}{N} \sum_{i=1}^N \frac{|x_i - \bar{x}|}{x_i}$

Вхідні дані

- Розглянемо дані компанії Apple в проміжку 01.01.1995 - 24.05.2020 з частотою спостережень в один тиждень

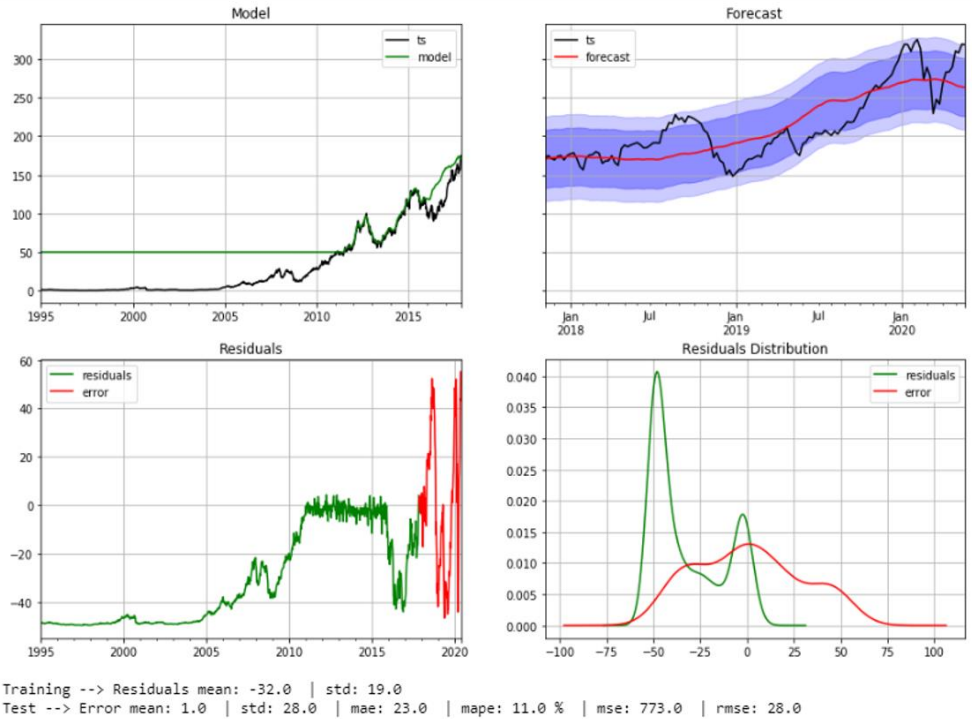


ARIMA



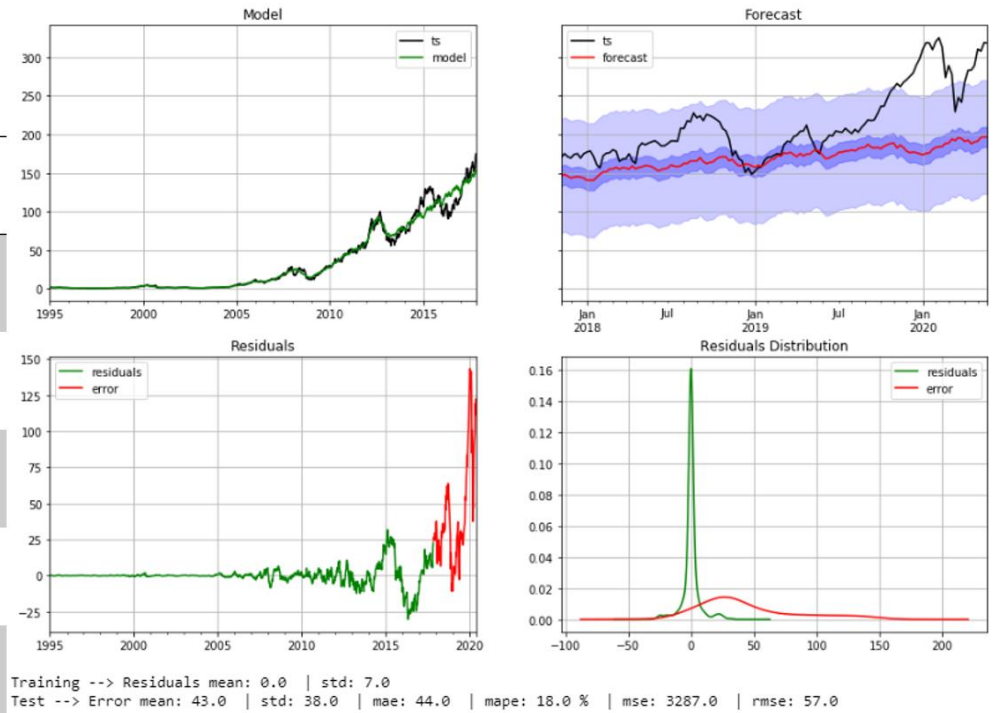
LSTM

KPI	Forecast
Error Mean	1
Error STD	28
MAE	23
MAPE	11%
MSE	773



PROPHET

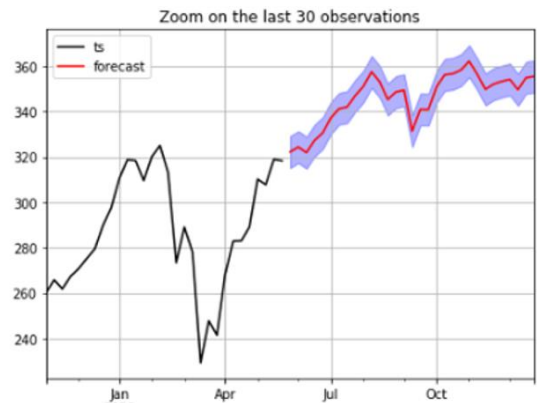
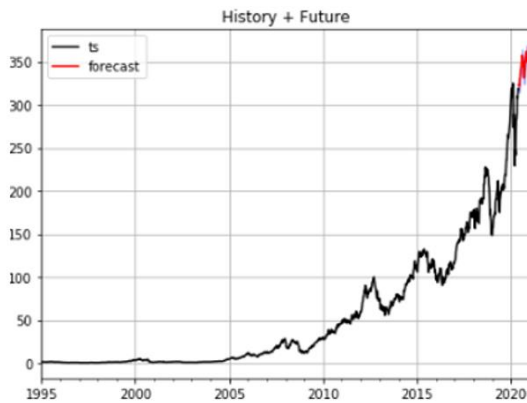
KPI	Forecast
Error Mean	43
Error STD	38
MAE	44
MAPE	18%
MSE	3287



Прогнозування на майбутній період

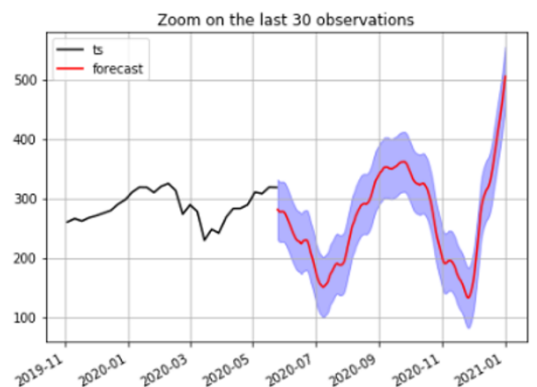
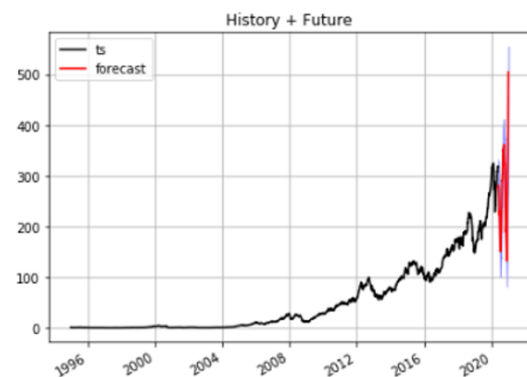
- ARIMA

-- generating index date --> start: 2020-05-31 00:00:00 | end: 2020-12-27 00:00:00 | len: 31 ---



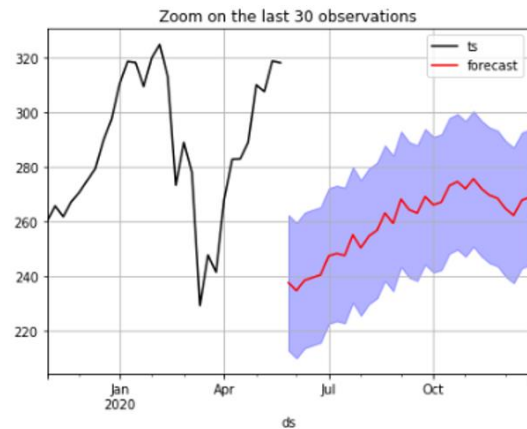
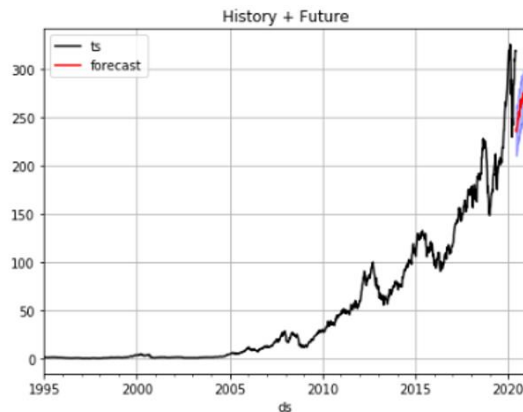
Прогнозування на майбутній період

- LSTM



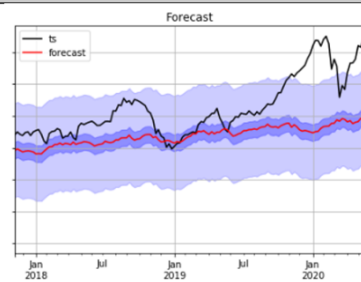
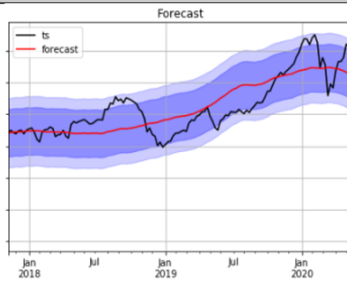
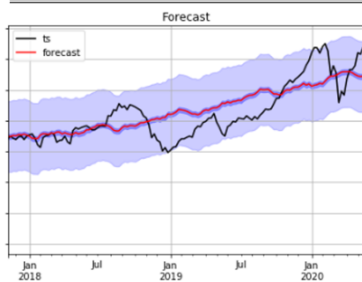
Прогнозування на майбутній період

- PROPHET



Аналіз результатів

KPI	ARIMA	LSTM	PROPHET
Error Mean	-5	1	43
Error STD	29	28	38
MAE	24	23	44
MAPE	12%	11%	18%
MSE	885	773	3287



Висновки

- Змодельовано часовий ряд вартості акцій компанії Apple
- Побудовані прогнози на тестовий та майбутній періоди часу
- Проведено порівняння методів ARIMA, LSTM та PROPHET
- Проаналізовано результати роботи методів

Шляхи подальшого розвитку

- Подальші дослідження за даною темою полягатимуть у детальнішому аналізі даних та експериментальному вибору найкращих параметрів для моделей
- Для більш точних прогнозів масив даних має бути більшим, а прогнози варто будувати на коротші періоди, для зменшення дисперсії
- Вдосконалити використання наведених моделей, та розглянути інші методи для вирішення даної задачі

ДОДАТОК Б ЛІСТИНГ ПРОГРАМИ

Лістинг розробленого програмного продукту:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import statsmodels.api as sm
from sklearn import preprocessing, svm
import pmdarima
import statsmodels.tsa.api as smt
import arch

from tensorflow.keras import models, layers, preprocessing as kprocessing
from fbprophet import Prophet
pd.plotting.register_matplotlib_converters()
from scipy import optimize

def plot_ts(ts, plot_ma=True, plot_intervals=True, window=30, figsize=(15,5)):
    rolling_mean = ts.rolling(window=window).mean()
    rolling_std = ts.rolling(window=window).std()
    plt.figure(figsize=figsize)
    plt.title(ts.name)
    plt.plot(ts[window:], label='Actual values', color="black")
    if plot_ma:
        plt.plot(rolling_mean, 'g', label='MA'+str(window), color="red")
    if plot_intervals:
        lower_bound = rolling_mean - (1.96 * rolling_std)
        upper_bound = rolling_mean + (1.96 * rolling_std)
        plt.fill_between(x=ts.index, y1=lower_bound, y2=upper_bound, color='lightskyblue', alpha=0.4)
    plt.legend(loc='best')
    plt.grid(True)
    plt.show()

def decompose_ts(ts, s=250, figsize=(20,13)):
    decomposition = smt.seasonal_decompose(ts, freq=s)
```

```

trend = decomposition.trend
seasonal = decomposition.seasonal
residual = decomposition.resid
fig, ax = plt.subplots(nrows=4, ncols=1, sharex=True, sharey=False, figsize=figsize)
ax[0].plot(ts)
ax[0].set_title('Original')
ax[0].grid(True)
ax[1].plot(trend)
ax[1].set_title('Trend')
ax[1].grid(True)
ax[2].plot(seasonal)
ax[2].set_title('Seasonality')
ax[2].grid(True)
ax[3].plot(residual)
ax[3].set_title('Residuals')
ax[3].grid(True)
return {"trend":trend, "seasonal":seasonal, "residual":residual}

def find_outliers(ts, perc=0.01, figsize=(15,5)):
    ## fit svm
    scaler = preprocessing.StandardScaler()
    ts_scaled = scaler.fit_transform(ts.values.reshape(-1,1))
    model = svm.OneClassSVM(nu=perc, kernel="rbf", gamma=0.01)
    model.fit(ts_scaled)
    ## dtf output
    dtf_outliers = ts.to_frame(name="ts")
    dtf_outliers["index"] = range(len(ts))
    dtf_outliers["outlier"] = model.predict(ts_scaled)
    dtf_outliers["outlier"] = dtf_outliers["outlier"].apply(lambda x: 1 if x==-1 else 0)
    ## plot
    fig, ax = plt.subplots(figsize=figsize)
    ax.set(title="Outliers detection: found "+str(sum(dtf_outliers["outlier"]==1)))
    ax.plot(dtf_outliers["index"], dtf_outliers["ts"], color="black")
    ax.scatter(x=dtf_outliers[dtf_outliers["outlier"]==1]["index"],
               y=dtf_outliers[dtf_outliers["outlier"]==1]["ts"],
               color='red')
    ax.grid(True)
    plt.show()

```



```
return dtf_outliers
```

```
def remove_outliers(ts, outliers_idx, figsize=(15,5)):
    ts_clean = ts.copy()
    ts_clean.loc[outliers_idx] = np.nan
    ts_clean = ts_clean.interpolate(method="linear")
    ax = ts.plot(figsize=figsize, color="red", alpha=0.5, title="Remove outliers", label="original", legend=True)
    ts_clean.plot(ax=ax, grid=True, color="black", label="interpolated", legend=True)
    plt.show()
    return ts_clean
```

```
def split_train_test(ts, exog=None, test=0.20, plot=True, figsize=(15,5)):

    if type(test) is float:
        split = int(len(ts)*(1-test))
        perc = test
    elif type(test) is str:
        split = ts.reset_index()[ts.reset_index().iloc[:,0]==test].index[0]
        perc = round(len(ts[split:])/len(ts), 2)
    else:
        split = test
        perc = round(len(ts[split:])/len(ts), 2)
    print("--- splitting at index: ", split, "|", ts.index[split], "| test size:", perc, " ---")
```

```
ts_train = ts.head(split)
ts_test = ts.tail(len(ts)-split)
if plot is True:
    fig, ax = plt.subplots(nrows=1, ncols=2, sharex=False, sharey=True, figsize=figsize)
    ts_train.plot(ax=ax[0], grid=True, title="Train", color="black")
    ts_test.plot(ax=ax[1], grid=True, title="Test", color="black")
    ax[0].set(xlabel=None)
    ax[1].set(xlabel=None)
    plt.show()
```

```
if exog is not None:
```

```

    exog_train = exog[0:split]
    exog_test = exog[split:]
    return ts_train, ts_test, exog_train, exog_test
else:
    return ts_train, ts_test

def utils_evaluate_forecast(dtf, title, plot=True, figsize=(20,13)):
    try:

        dtf["residuals"] = dtf["ts"] - dtf["model"]
        dtf["error"] = dtf["ts"] - dtf["forecast"]
        dtf["error_pct"] = dtf["error"] / dtf["ts"]

        residuals_mean = dtf["residuals"].mean()
        residuals_std = dtf["residuals"].std()
        error_mean = dtf["error"].mean()
        error_std = dtf["error"].std()
        mae = dtf["error"].apply(lambda x: np.abs(x)).mean()
        mape = dtf["error_pct"].apply(lambda x: np.abs(x)).mean()
        mse = dtf["error"].apply(lambda x: x**2).mean()
        rmse = np.sqrt(mse)

        dtf["conf_int_low"] = dtf["forecast"] - 1.96*residuals_std
        dtf["conf_int_up"] = dtf["forecast"] + 1.96*residuals_std
        dtf["pred_int_low"] = dtf["forecast"] - 1.96*error_std
        dtf["pred_int_up"] = dtf["forecast"] + 1.96*error_std

    if plot==True:
        fig = plt.figure(figsize=figsize)
        fig.suptitle(title, fontsize=20)
        ax1 = fig.add_subplot(2,2, 1)
        ax2 = fig.add_subplot(2,2, 2, sharey=ax1)
        ax3 = fig.add_subplot(2,2, 3)
        ax4 = fig.add_subplot(2,2, 4)

```

```

dtf[pd.notnull(dtf["model"])]["ts","model"].plot(color=["black","green"], title="Model", grid=True, ax=ax1)
ax1.set(xlabel=None)

dtf[pd.isnull(dtf["model"])]["ts","forecast"].plot(color=["black","red"], title="Forecast", grid=True, ax=ax2)
ax2.fill_between(x=dtf.index, y1=dtf['pred_int_low'], y2=dtf['pred_int_up'], color='b', alpha=0.2)
ax2.fill_between(x=dtf.index, y1=dtf['conf_int_low'], y2=dtf['conf_int_up'], color='b', alpha=0.3)
ax2.set(xlabel=None)

dtf[["residuals","error"]].plot(ax=ax3, color=["green","red"], title="Residuals", grid=True)
ax3.set(xlabel=None)

dtf[["residuals","error"]].plot(ax=ax4, color=["green","red"], kind='kde', title="Residuals Distribution",
grid=True)
ax4.set(ylabel=None)
plt.show()
print("Training --> Residuals mean:", np.round(residuals_mean), " | std:", np.round(residuals_std))
print("Test --> Error mean:", np.round(error_mean), " | std:", np.round(error_std),
      " | mae:", np.round(mae), " | mape:", np.round(mape*100), "% | mse:", np.round(mse), " |
rmse:", np.round(rmse))

return dtf[["ts","model","residuals","conf_int_low","conf_int_up",
            "forecast","error","pred_int_low","pred_int_up"]]

except Exception as e:
    print("--- got error ---")
    print(e)

def utils_generate_indexdate(start, end=None, n=None, freq="D"):
    if end is not None:
        index = pd.date_range(start=start, end=end, freq=freq)
    else:
        index = pd.date_range(start=start, periods=n, freq=freq)
    index = index[1:]
    print("--- generating index date --> start:", index[0], " | end:", index[-1], " | len:", len(index), "---")
    return index

```

```

def utils_plot_forecast(dtf, zoom=30, figsize=(15,5)):
    ## interval
    dtf["residuals"] = dtf["ts"] - dtf["model"]
    dtf["conf_int_low"] = dtf["forecast"] - 1.96*dtf["residuals"].std()
    dtf["conf_int_up"] = dtf["forecast"] + 1.96*dtf["residuals"].std()
    fig, ax = plt.subplots(nrows=1, ncols=2, figsize=figsize)

    ## entire series
    dtf[["ts", "forecast"]].plot(color=["black", "red"], grid=True, ax=ax[0], title="History + Future")
    ax[0].fill_between(x=dtf.index, y1=dtf['conf_int_low'], y2=dtf['conf_int_up'], color='b', alpha=0.3)

    ## focus on last
    first_idx = dtf[pd.notnull(dtf["forecast"])].index[0]
    first_loc = dtf.index.tolist().index(first_idx)
    zoom_idx = dtf.index[first_loc-zoom]
    dtf.loc[zoom_idx:][["ts", "forecast"]].plot(color=["black", "red"], grid=True, ax=ax[1], title="Zoom on the last
"+str(zoom)+" observations")
    ax[1].fill_between(x=dtf.loc[zoom_idx:].index, y1=dtf.loc[zoom_idx:]['conf_int_low'],
                      y2=dtf.loc[zoom_idx:]['conf_int_up'], color='b', alpha=0.3)
    plt.show()
    return dtf[["ts", "model", "residuals", "conf_int_low", "forecast", "conf_int_up"]]

def fit_expsmooth(ts_train, ts_test, trend="additive", seasonal="multiplicative", s=None, alpha=0.94, figsize=(15,10)):

    check_seasonality = "Seasonal parameters: No Seasonality" if (seasonal is None) & (s is None) else "Seasonal
parameters: "+str(seasonal)+" Seasonality every "+str(s)+" observations"
    print(check_seasonality)

    model = smt.ExponentialSmoothing(ts_train, trend=trend, seasonal=seasonal,
seasonal_periods=s).fit(smoothing_level=alpha)
    dtf_train = ts_train.to_frame(name="ts")
    dtf_train["model"] = model.fittedvalues

```

```

dtf_test = ts_test.to_frame(name="ts")
dtf_test["forecast"] = model.predict(start=len(ts_train), end=len(ts_train)+len(ts_test)-1)

dtf = dtf_train.append(dtf_test)
dtf = utils_evaluate_forecast(dtf, figsize=figsize, title="Holt-Winters (" + str(alpha) + ")")
return dtf, model

def fit_sarimax(ts_train, ts_test, order=(1,0,1), seasonal_order=(0,0,0,0), exog_train=None, exog_test=None,
figsize=(15,10)):

    check_trend = "Trend parameters: No differencing" if order[1] == 0 else "Trend parameters: d=" + str(order[1])
    print(check_trend)
    check_seasonality = "Seasonal parameters: No Seasonality" if (seasonal_order[3] == 0) &
(np.sum(seasonal_order[0:3]) == 0) else "Seasonal parameters: Seasonality every " + str(seasonal_order[3]) + "
observations"
    print(check_seasonality)
    check_exog = "Exog parameters: Not given" if (exog_train is None) & (exog_test is None) else "Exog parameters:
number of regressors=" + str(exog_train.shape[1])
    print(check_exog)

    model = smt.SARIMAX(ts_train, order=order, seasonal_order=seasonal_order, exog=exog_train,
enforce_stationarity=False, enforce_invertibility=False).fit()
    dtf_train = ts_train.to_frame(name="ts")
    dtf_train["model"] = model.fittedvalues

    dtf_test = ts_test.to_frame(name="ts")
    dtf_test["forecast"] = model.predict(start=len(ts_train), end=len(ts_train)+len(ts_test)-1, exog=exog_test)

    dtf = dtf_train.append(dtf_test)
    title = "ARIMA " + str(order) if exog_train is None else "ARIMAX " + str(order)
    title = "S" + title + " x " + str(seasonal_order) if np.sum(seasonal_order) > 0 else title
    dtf = utils_evaluate_forecast(dtf, figsize=figsize, title=title)

```

```

return dtf, model

def find_best_sarimax(ts, seasonal=True, stationary=False, s=1, exog=None,
                    max_p=10, max_d=3, max_q=10,
                    max_P=10, max_D=3, max_Q=10):
    best_model = pmdarima.auto_arima(ts, exogenous=exog,
                                    seasonal=seasonal, stationary=stationary, m=s,
                                    information_criterion='aic', max_order=20,
                                    max_p=max_p, max_d=max_d, max_q=max_q,
                                    max_P=max_P, max_D=max_D, max_Q=max_Q,
                                    error_action='ignore')
    print("best model --> (p, d, q):", best_model.order, " and (P, D, Q, s):", best_model.seasonal_order)
    return best_model.summary()

def fit_garch(ts_train, ts_test, order=(1,0,1), seasonal_order=(0,0,0,0), exog_train=None, exog_test=None,
             figsize=(15,10)):
    ## train
    arima = smt.SARIMAX(ts_train, order=order, seasonal_order=seasonal_order, exog=exog_train,
                       enforce_stationarity=False, enforce_invertibility=False).fit()
    garch = arch.arch_model(arima.resid, p=order[0], o=order[1], q=order[2], x=exog_train, dist='StudentsT', power=2.0,
                           mean='Constant', vol='GARCH')
    model = garch.fit(update_freq=seasonal_order[3])
    dtf_train = ts_train.to_frame(name="ts")
    dtf_train["model"] = model.conditional_volatility

    ## test
    dtf_test = ts_test.to_frame(name="ts")
    dtf_test["forecast"] = model.forecast(horizon=len(ts_test))

    ## evaluate
    dtf = dtf_train.append(dtf_test)
    title = "GARCH (" + str(order[0]) + ", " + str(order[2]) + ")" if order[0] != 0 else "ARCH (" + str(order[2]) + ")"
    dtf = utils_evaluate_forecast(dtf, figsize=figsize, title=title)
    return dtf, model

def forecast_arima(ts, model, pred_ahead=None, end=None, freq="D", zoom=30, figsize=(15,5)):
    ## fit

```

```

model = model.fit()
dtf = ts.to_frame(name="ts")
dtf["model"] = model.fittedvalues
dtf["residuals"] = dtf["ts"] - dtf["model"]

index = utils_generate_indexdate(start=ts.index[-1], end=end, n=pred_ahead, freq=freq)

preds = model.forecast(len(index))
dtf = dtf.append(preds.to_frame(name="forecast"))

dtf = utils_plot_forecast(dtf, zoom=zoom)
return dtf

def utils_plot_keras_training(training):
    metrics = [k for k in training.history.keys() if ("loss" not in k) and ("val" not in k)]
    fig, ax = plt.subplots(nrows=1, ncols=2, sharey=True, figsize=(15,3))

    ## training
    ax[0].set(title="Training")
    ax11 = ax[0].twinx()
    ax[0].plot(training.history['loss'], color='black')
    ax[0].set_xlabel('Epochs')
    ax[0].set_ylabel('Loss', color='black')
    for metric in metrics:
        ax11.plot(training.history[metric], label=metric)
    ax11.set_ylabel("Score", color='steelblue')
    ax11.legend()

    ## validation
    ax[1].set(title="Validation")
    ax22 = ax[1].twinx()
    ax[1].plot(training.history['val_loss'], color='black')
    ax[1].set_xlabel('Epochs')
    ax[1].set_ylabel('Loss', color='black')

```

```

for metric in metrics:
    ax22.plot(training.history['val_'+metric], label=metric)
ax22.set_ylabel("Score", color="steelblue")
plt.show()

def utils_preprocess_ts(ts, s, scaler=None, exog=None):
    ## scale
    if scaler is None:
        scaler = preprocessing.MinMaxScaler(feature_range=(0,1))
    ts_preprocessed = scaler.fit_transform(ts.values.reshape(-1,1)).reshape(-1)

    ## create X,y for train
    ts_preprocessed = kprocessing.sequence.TimeseriesGenerator(data=ts_preprocessed,
                                                                targets=ts_preprocessed,
                                                                length=s, batch_size=1)

    lst_X, lst_y = [], []
    for i in range(len(ts_preprocessed)):
        xi, yi = ts_preprocessed[i]
        lst_X.append(xi)
        lst_y.append(yi)
    X = np.array(lst_X)
    y = np.array(lst_y)
    return X, y, scaler

def utils_fitted_lstm(ts, model, scaler, exog=None):
    ## scale
    ts_preprocessed = scaler.fit_transform(ts.values.reshape(-1,1)).reshape(-1)

    ## create Xy, predict = fitted
    s = model.input_shape[-1]
    lst_fitted = [np.nan]*s
    for i in range(len(ts_preprocessed)):
        end_ix = i + s
        if end_ix > len(ts_preprocessed)-1:
            break

```



```

    X = ts_preprocessed[i:end_ix]
    X = np.array(X)
    X = np.reshape(X, (1,1,X.shape[0]))
    fit = model.predict(X)
    fit = scaler.inverse_transform(fit)[0][0]
    lst_fitted.append(fit)
return np.array(lst_fitted)

def utils_predict_lstm(ts, model, scaler, pred_ahead, exog=None):
    ## scale
    s = model.input_shape[-1]
    ts_preprocessed = list(scaler.fit_transform(ts[-s:].values.reshape(-1,1)))

    ## predict, append, re-predict
    lst_preds = []
    for i in range(pred_ahead):
        X = np.array(ts_preprocessed[len(ts_preprocessed)-s:])
        X = np.reshape(X, (1,1,X.shape[0]))
        pred = model.predict(X)
        ts_preprocessed.append(pred)
        pred = scaler.inverse_transform(pred)[0][0]
        lst_preds.append(pred)
    return np.array(lst_preds)

def fit_lstm(ts_train, ts_test, model, exog=None, s=20, figsize=(15,5)):
    ## check
    print("Seasonality: using the last", s, "observations to predict the next 1")

    ## preprocess train
    X_train, y_train, scaler = utils_preprocess_ts(ts_train, scaler=None, exog=exog, s=s)

    ## lstm
    if model is None:
        model = models.Sequential()
        model.add( layers.LSTM(input_shape=X_train.shape[1:], units=50, activation='relu', return_sequences=False) )
        model.add( layers.Dense(1) )

```

```

model.compile(optimizer='adam', loss='mean_absolute_error')

## train
print(model.summary())
training = model.fit(x=X_train, y=y_train, batch_size=1, epochs=100, shuffle=True, verbose=0, validation_split=0.3)
utils_plot_keras_training(training)

dtf_train = ts_train.to_frame(name="ts")
dtf_train["model"] = utils_fitted_lstm(ts_train, training.model, scaler, exog)
dtf_train["model"] = dtf_train["model"].fillna(method='bfill')

## test
preds = utils_predict_lstm(ts_train[-s:], training.model, scaler, pred_ahead=len(ts_test), exog=None)
dtf_test = ts_test.to_frame(name="ts").merge(pd.DataFrame(data=preds, index=ts_test.index, columns=["forecast"]),
                                           how='left', left_index=True, right_index=True)

## evaluate
dtf = dtf_train.append(dtf_test)
dtf = utils_evaluate_forecast(dtf, figsize=figsize, title="LSTM (memory:"+str(s)+"")
return dtf, training.model

def forecast_lstm(ts, model, pred_ahead=None, end=None, freq="D", zoom=30, figsize=(15,5)):
    ## fit
    s = model.input_shape[-1]
    X, y, scaler = utils_preprocess_ts(ts, scaler=None, exog=None, s=s)
    training = model.fit(x=X, y=y, batch_size=1, epochs=100, shuffle=True, verbose=0, validation_split=0.3)
    dtf = ts.to_frame(name="ts")
    dtf["model"] = utils_fitted_lstm(ts, training.model, scaler, None)
    dtf["model"] = dtf["model"].fillna(method='bfill')

    ## index
    index = utils_generate_indexdate(start=ts.index[-1], end=end, n=pred_ahead, freq=freq)

    ## forecast
    preds = utils_predict_lstm(ts[-s:], training.model, scaler, pred_ahead=len(index), exog=None)
    dtf = dtf.append(pd.DataFrame(data=preds, index=index, columns=["forecast"]))

```

```

## plot
dtf = utils_plot_forecast(dtf, zoom=zoom)
return dtf

def fit_prophet(dtf_train, dtf_test, lst_exog=None, model=None, freq="D", figsize=(15,10)):
    ## setup prophet
    if model is None:
        model = Prophet(growth="linear", changepoints=None, n_changepoints=25, seasonality_mode="multiplicative",
                        yearly_seasonality="auto", weekly_seasonality="auto", daily_seasonality="auto",
                        holidays=None)
    if lst_exog != None:
        for regressor in lst_exog:
            model.add_regressor(regressor)

    ## train
    model.fit(dtf_train)

    ## test
    dtf_prophet = model.make_future_dataframe(periods=len(dtf_test), freq=freq, include_history=True)

    if model.growth == "logistic":
        dtf_prophet["cap"] = dtf_train["cap"].unique()[0]

    if lst_exog != None:
        dtf_prophet = dtf_prophet.merge(dtf_train[["ds"]+lst_exog], how="left")
        dtf_prophet.iloc[-len(dtf_test):][lst_exog] = dtf_test[lst_exog].values

    dtf_prophet = model.predict(dtf_prophet)
    dtf_train      = dtf_train.merge(dtf_prophet[["ds", "yhat"]],      how="left").rename(columns={'yhat': 'model',
'y': 'ts'}).set_index("ds")
    dtf_test       = dtf_test.merge(dtf_prophet[["ds", "yhat"]],      how="left").rename(columns={'yhat': 'forecast',
'y': 'ts'}).set_index("ds")

    ## evaluate
    dtf = dtf_train.append(dtf_test)
    dtf = utils_evaluate_forecast(dtf, figsize=figsize, title="Prophet")
    return dtf, model

```

```

def forecast_prophet(dtf, model, pred_ahead=None, end=None, freq="D", zoom=30, figsize=(15,5)):
    ## fit
    model.fit(dtf)

    ## index
    index = utils_generate_indexdate(start=dtf["ds"].values[-1], end=end, n=pred_ahead, freq=freq)

    ## forecast
    dtf_prophet = model.make_future_dataframe(periods=len(index), freq=freq, include_history=True)
    dtf_prophet = model.predict(dtf_prophet)
    dtf = dtf.merge(dtf_prophet[["ds", "yhat"]], how="left").rename(columns={'yhat': 'model', 'y': 'ts'}).set_index("ds")
    preds = pd.DataFrame(data=index, columns=["ds"])
    preds = preds.merge(dtf_prophet[["ds", "yhat"]], how="left").rename(columns={'yhat': 'forecast'}).set_index("ds")
    dtf = dtf.append(preds)

    ## plot
    dtf = utils_plot_forecast(dtf, zoom=zoom)
    return dtf

```

Порівняльний аналіз методів ARIMA, LSTM, PROPHET:

```

import warnings
warnings.filterwarnings("ignore")
from ts_utils import *

dtf = pd.read_csv('AAPLWeek.csv') #TM Historical Data.csv
dtf = dtf.iloc[::1]
dtf = dtf.set_index("Date")
dtf.tail()

dtf1 = pd.read_csv('AAPLDay.csv') #TM Historical Data.csv
dtf1 = dtf1.set_index("Date")
dtf1["Date"] = pd.to_datetime(dtf1.index, format='%m/%d/%Y')
dtf.head()

dtf["date"] = pd.to_datetime(dtf.index, format='%m/%d/%Y')

```

```

ts = dtf.groupby("date")["Close"].sum().rename("Stock Price $")
ts.head()

plt.figure(figsize=(15,5))
plt.grid(True)
plt.title(ts.name)
plt.plot(ts,color="red",label='Actual values', marker='o')

print("population --> len:", len(ts), " | mean:", round(ts.mean()), " | std:", round(ts.std()))
w = 13
print("moving --> len:", w, " | mean:", round(ts.ewm(span=w).mean()[-1]), " | std:", round(ts.ewm(span=w).std()[-1]))

plot_ts(ts, plot_ma=True, plot_intervals=True, window=13, figsize=(15,5))

plot_ts(ts, plot_ma=True, plot_intervals=True, window=52, figsize=(15,5))

dtf_outliers = find_outliers(ts, perc=0.05, figsize=(15,5))

ts_clean = remove_outliers(ts, outliers_idx=dtf_outliers[dtf_outliers["outlier"]==1].index, figsize=(15,5))

dic_decomposed = decompose_ts(ts, s=30, figsize=(15,10))

ts_train, ts_test = split_train_test(ts, exog=None, test=0.1, plot=True, figsize=(15,5))

find_best_sarimax(ts, seasonal=True, stationary=False, s=s, exog=None,
                  max_p=10, max_d=5, max_q=10,
                  max_P=5, max_D=5, max_Q=5)

dtf, model_arima_fitted = fit_sarimax(ts_train, ts_test, order=(3,2,1), seasonal_order=(3,2,1,s), figsize=(15,10)) #
seasonal_order=(2,2,1,26),

model_arima = smt.SARIMAX(ts, order=(3,2,1), seasonal_order=(3,2,1,s), exog=None)

future = forecast_arima(ts, model_arima, end="2021-01-01",freq="W", zoom=30, figsize=(15,5))

model_lstm = models.Sequential()
model_lstm.add(layers.LSTM(input_shape=(1,s2), units=50, activation='relu', return_sequences=False) )

```

```

model_lstm.add( layers.Dense(1) )
model_lstm.compile(optimizer='adam', loss='mean_absolute_error')

ts_train_arr = np.asarray(ts_train,dtype=np.float64)
ts_test_arr = np.asarray(ts_test, dtype=np.float64)

ts_train_arr = np.expand_dims(ts_train_arr, -1)
ts_test_arr = np.expand_dims(ts_test_arr, -1)

dtf, model_lstm_fited = fit_lstm(ts_train, ts_test, model_lstm, exog=None, s=s2, figsize=(15,10))

future = forecast_lstm(ts, model_lstm_fited, end="2021-01-01", freq="D", zoom=30, figsize=(15,5))

dtf_train = ts_train.reset_index().rename(columns={"date":"ds", "Stock Price $":"y"})
dtf_test = ts_test.reset_index().rename(columns={"date":"ds", "Stock Price $":"y"})

dtf_train.tail()

model_prophet = Prophet(changepoint_prior_scale=30, seasonality_prior_scale=35,#holidays__prior_scale=30,
                        seasonality_mode="multiplicative",
                        yearly_seasonality=False, weekly_seasonality=False).add_seasonality(
                        name='mounthly',
                        period=4.5,
                        fourier_order=12).add_seasonality(
                        name='weekly',
                        period=1,
                        fourier_order=20).add_seasonality(
                        name='yearly',
                        period=52,
                        fourier_order=60).add_seasonality(
                        name='quarterly',
                        period=13, fourier_order=5,prior_scale = 15)

dtf, model_prophet = fit_prophet(dtf_train, dtf_test, model=model_prophet, freq="W", figsize=(15,10))

model_prophet1 = Prophet()

dtf, model_prophet = fit_prophet(dtf_train, dtf_test, model=model_prophet1, freq="W", figsize=(15,10))

```

```
dtf = ts.reset_index().rename(columns={"date":"ds", "Stock Price $":"y"})
dtf.tail()
```

```
model_prophet = Prophet(changepoint_prior_scale=30, seasonality_prior_scale=35, #holidays__prior_scale=30,
                        seasonality_mode="multiplicative",
                        yearly_seasonality=False, weekly_seasonality=False).add_seasonality(
                        name='mounthly',
                        period=4.5,
                        fourier_order=12).add_seasonality(
                        name='weekly',
                        period=1,
                        fourier_order=20).add_seasonality(
                        name='yearly',
                        period=52,
                        fourier_order=100).add_seasonality(
                        name='quarterly',
                        period=13, fourier_order=5, prior_scale = 15)
```

```
future = forecast_prophet(dtf, model_prophet, end="2021-01-01", freq="W", zoom=30, figsize=(15,5))
```